NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

EVALUATION OF OBJECT DETECTION ALGORITHMS FOR SHIP DETECTION IN THE VISIBLE SPECTRUM

by

David M. Camp

December 2013

Thesis Co-Advisors: Douglas Fouts
Mathias Kölsch

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2013	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE EVALUATION OF OBJECT DETECTION ALGORITHMS FOR SHIP DETECTION IN THE VISIBLE SPECTRUM	**5. FUNDING NUMBERS**
6. AUTHOR(S) David M. Camp	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. government. IRB protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

The research described here examined computer vision algorithms for suitability to aid or replace the current methods of ship detection and tracking from a photonics mast. Evaluation was conducted on three object detection methods: a bag of words (BOW) robust multi-class classification method; a histogram of oriented gradient (HOG) method, originally used for pedestrian tracking; and a deformable parts model (DPM) that was originally designed for pose recognition that has been successful in multi-class classification. A fourth method that combines the HOG and BOW was created and successfully reduced false positive detections while maintaining a high recall rate.

The object detection methods were evaluated through a search theory model to frame evaluation for operational ship detection. Each object detection method was optimized following a design of experiments approach utilizing a cluster computer. The BOW method had the highest recall for ships 25 pixels and smaller, while the HOG method was the fastest of all methods when implemented on a graphical processing unit. The DPM method had the highest average recall for ships greater than 25 pixels but the lowest recall for smaller ships. Finally, the hybrid HOG and BOW method had the highest mean recall and lowest mean false positive rate over all ship sizes.

14. SUBJECT TERMS Ship detection, search theory, computer vision, cluster computer, graphical processor unit (GPU), parallel computing, design of experiments, object detection	**15. NUMBER OF PAGES** 141
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

EVALUATION OF OBJECT DETECTION ALGORITHMS FOR SHIP DETECTION IN THE VISIBLE SPECTRUM

David M. Camp
Lieutenant, United States Navy
B.S. University of Washington, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 2013

Author: David M. Camp

Approved by: Douglas Fouts
Thesis Co-Advisor

Mathias Kölsch
Thesis Co-Advisor

Clark Robertson
Chair, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The research described here examined computer vision algorithms for suitability to aid or replace the current methods of ship detection and tracking from a photonics mast. Evaluation was conducted on three object detection methods: a bag of words (BOW) robust multi-class classification method; a histogram of oriented gradient (HOG) method, originally used for pedestrian tracking; and a deformable parts model (DPM) that was originally designed for pose recognition that has been successful in multi-class classification. A fourth method that combines the HOG and BOW was created and successfully reduced false positive detections while maintaining a high recall rate.

The object detection methods were analyzed through a search theory model to frame evaluation for operational ship detection. Each object detection method was optimized following a design of experiments approach utilizing a cluster computer. The BOW method had the highest recall for ships 25 pixels and smaller, while the HOG method was the fastest of all methods when implemented on a graphical processing unit. The DPM method had the highest average recall for ships greater than 25 pixels but the lowest recall for smaller ships. Finally, the hybrid HOG and BOW method had the highest mean recall and lowest mean false positive rate over all ship sizes.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

1-precissoin	False positive detection percentage
δ	Glimpse probability
ω	Sweep width
AP%	Average precision percentage
BOW	Bag of Words
BRIEF	Binary robust independent elementary features
BRISK	Binary robust invariant scalable keypoints
CenSurE (STAR)	Center surround extrema
CPA	Closest point of approach
CPU	Central processing unit
DDR	Double data rate
DoG	Difference of Gaussian
DPM	Deformable parts model
FAST	Feature from accelerated segment test
f_l	Focal length
FLANN	Fast library approximate of nearest neighbors
FOV	Field of view
fps	Frames per second
GB	Gigabyte
GFTT	Good features to track
GPU	Graphical processing unit
HARRIS	Harris corner detector
HOG	Histogram of oriented gradients
HYBRID	HOG and BOW hybrid detection method
ISR	Intelligence, surveillance and recognizance
kB	Kilobyte
L1	Brute force Manhattan distance
L2	Brute force Euclidean distance
MB	Megabyte
MFOV	Medium field of view

MSER	Maximally stable extremal region
MHH	Masthead height
NOAA	National Oceanic and Atmospheric Association
NFOV	Narrow field of view
OpenCL	Open standard for parallel programming of heterogeneous systems
OpenCV	Open source computer vision
OpenMP	Open source high performance computing
ORB	Oriented fast and rotated BRIEF
PASCAL	Pattern analysis, statistical modeling and computational learning
PCI	Peripheral component interconnect
P_D	Probability of detection
RADAR	Radio detection and ranging
RAM	Random access memory
Recall	True positive detection percentage
RGB	Red, green, blue
ROC	Receiver operator characteristics
ROI	Region of interest
SATA	Serial advanced technology attachment
SIFT	Scale-invariant feature transform
SONAR	Sound navigation and ranging
SSD	Solid-state drive
SURF	Speeded up robust features
SVM	Support vector machine
TBB	Thread building blocks
t_f	Frame time
t_s	Sweep time
UNFOV	Ultra narrow field of view
VOC	Visual object classes challenge
WFOV	Wide field of view

EXECUTIVE SUMMARY

The U.S. Navy, like many organizations, continues to have computers automate tasks that were previously accomplished by people. It is only beneficial to have a computer automate a task if the computer can perform the task as well as an operator. The challenges of using computers to perform automatic visual ship detection on a submarine were investigated in this thesis, and how well a computer could perform this task was considered. There are many automated tools to help detect and track ships from a submarine, most of which are for sound navigation and ranging (SONAR) and radio detection and ranging (RADAR) systems. Operator understanding of the tactical and contact picture can be improved by automatic visual ship detection. One benefit of visual ship detection is that it operates passively, requiring no transmission. Additionally, it can provide very accurate bearing and range information. The visual sensor can also detect ships that may not be detected by SONAR and RADAR.

A search theory model used for the evaluation and comparison of SONAR and RADAR systems was adapted for this investigation [1]. The evaluation model was used to incorporate many of the challenges faced by a submarine in visual ship detection. These challenges included detecting ships at great distances in open seas and detecting ships in the midst of a harbor. Incorporating these challenges into the evaluation model provided a better understanding of the capabilities of the computer vision algorithms investigated. Many computer vision algorithms have been utilized for visual ship detection, and many of these algorithms are considered. Most of these approaches were found to have limitations if used onboard a submarine. Some of these limitations include using frame differences or blocks above the horizon, to be considered as ships. Instead of considering these methods, object detection methods were examined that have been successful for many classes of objects.

Three-object detection algorithms were evaluated for use as a visual ship detector onboard a submarine. The first method was a bag of words (BOW) approach, a robust multiclass object detection method that allows the evaluation of many computer vision visual feature methods [2]. In the BOW evaluation, 10 methods of selecting visual

feature keypoints were investigated. The second method investigated was a histogram of oriented gradients (HOG) approach that has been very successful in tracking pedestrians [3]. The third method was a deformable parts model (DPM), which has been used for detecting many types of objects and can detect and differentiate people performing different poses [4]. Lastly, a hybrid ship detector was created by combining the HOG and BOW methods (HYBRID). The HYBRID detector provides the benefits of speed of detection and localization from HOG while successfully reducing the false positive rate to the lowest levels of all evaluated detectors. The HYBRID method is capable of being trained to utilize the multiclass detection of the BOW method, allowing it to differentiate between classes of ships such as a merchant or a warship.

To create the best ship detector from each of the three object detection methods, a design of experiments was conducted to train the detectors on a cluster computer. From the design of experiments, thousands of ship detectors were created from each object detection method. The training parameters that produced the best ship detectors are described for each object detection method. Out of the thousands of ship detectors, the top detectors from each method were evaluated using the search theory evaluation model. Then, the top detectors from each method were compared against each other and against the HYBRID model. Using the search theory evaluation model provided an expectation for the operational use of each ship detector without actually performing an operational test. The evaluation takes into account the probability of detection per glimpse of the target by range and the time between possible glimpses while performing 360-degree scans with the detector.

Scaled ship images are shown in Figure 1, with an initial average size of 256 pixels tall and scaled to 75, 50, 25, 20, 15 and 10 percent of the original image size. The ship sizes in Figure 1 are similar to the sizes used to simulate the detection of ships at greater ranges. The HYBRID detector had the best average results over all scales with an 89.14 percent detection rate and a 10.28 percent false positive rate. The DPM ship detector was second with average rates over all scales of 85.57 percent and an 11 percent false positive rate, followed by HOG and then BOW. Additionally, the HOG was the fastest computationally when performed on a GPU at greater than eight, 1920 by 1080 pixel, frames per second (fps), followed by BOW and HYBRID at five and two fps. Finally, the DPM method was the slowest at one frame every two seconds, though faster implementations of DPM are discussed and considered through the model in the evaluation.

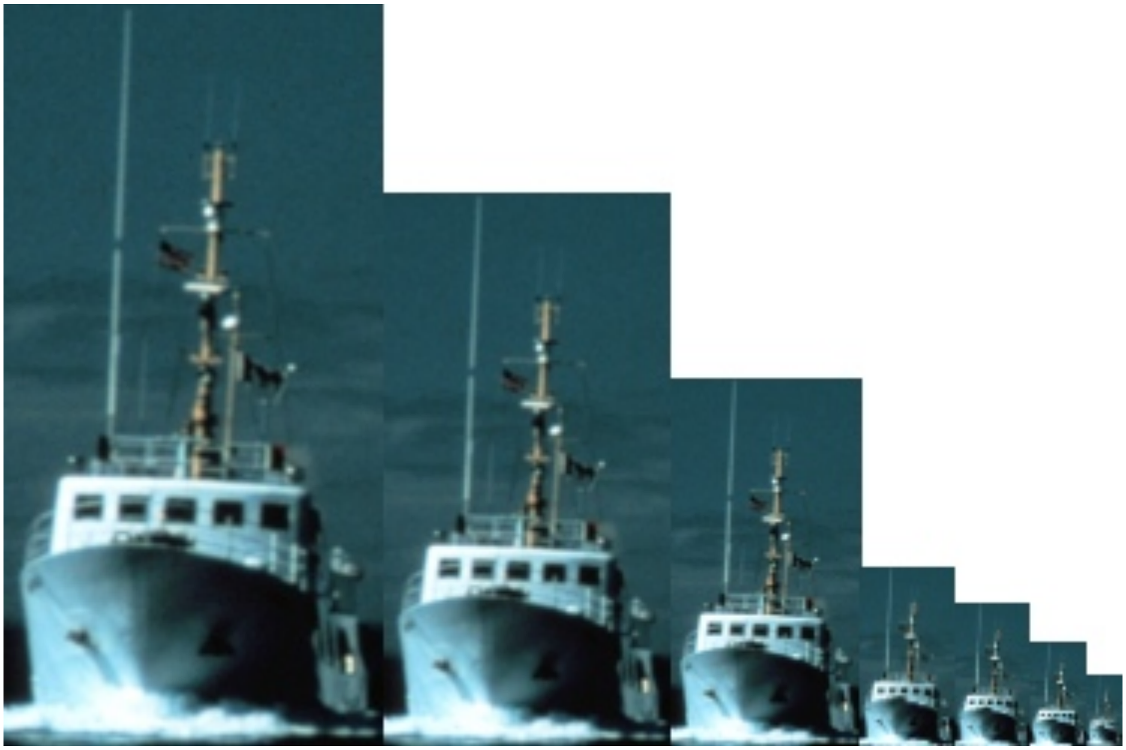

Figure 1. Example of a cropped ship image with the average pixel height of the evaluation set, and the subsequent 75, 50, 25, 20, 15 and 10 percent scales of the image, used to find probability of detection by range (after [5]).

The results of using the search theory evaluation model provided the expectation that the HYBRID detector would detect a 10-meter masthead height (MHH) contact traveling at a relative speed of 20 knots at a 9.7-kilometer sweep width. Simply put, the HYRID model should detect this contact by the time it is within 9.7-kilometers of the sensor. This was given that the sensor, lens, focal length and scan techniques used in the evaluation model are followed, to include a sensor being only two meters above sea level. Using these evaluation restrictions, we achieved expectations for the detection range (sweep width). The sweep width for a 20 knots relative speed contact by MHH is illustrated in Figure 2 for the top ship detector from each model evaluated. The rest of this thesis is a description in detail how the evaluation model was created and how these object detection methods were selected and adapted to be ship detectors, along with their individual performances.

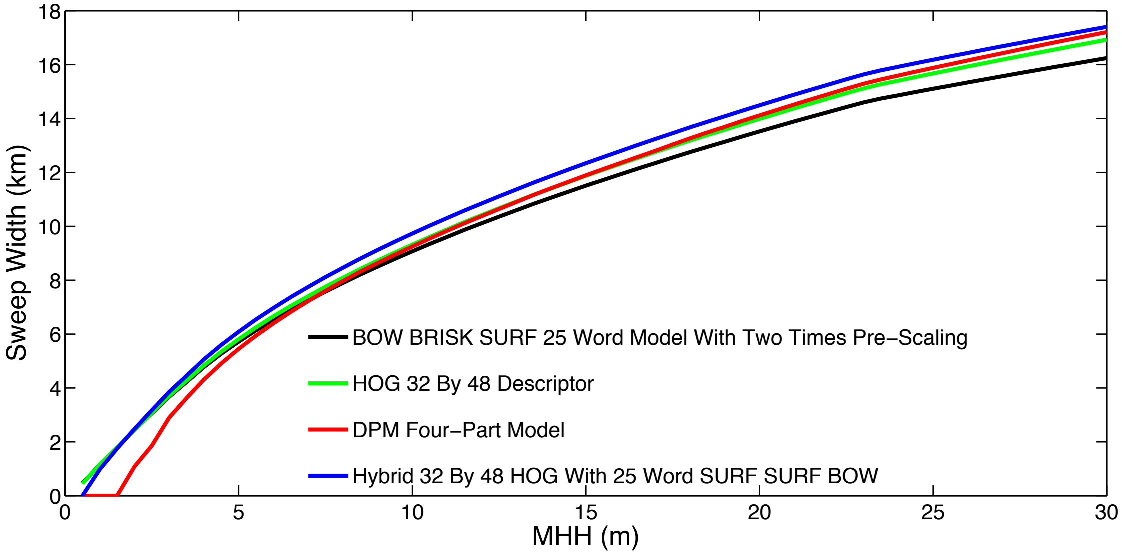

Figure 2. Expected detection range (sweep width) of the best ship detector from each method for a 20 knot relative velocity contact based on the MHH.

LIST OF REFERENCES

[1] D. H. Wagner et al., Eds., *Naval Operations Analysis* 3rd ed. Annapolis, MD: Naval Institute Press, 1999.

[2] G. Csurka et al., “Visual Categoriation With Bags of Keypoints,” in *Workshop on Statistical Learning in Computer Vision, ECCV*, Prague, Czech Republic, 2004, pp. 1–22.

[3] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, San Diego, CA, 2005, pp. 886–893.

[4] Y. Yang and D. Ramanan, “Articulated pose estimation with flexible mixtures-of-parts,” in *Conference on Computer Vision and Pattern Recognition*, Colorado Srpings, CO, 2011, pp. 1385–1392.

[5] National Oceanic and Atmoshperic Administration (2012). National Oceanic and Atmoshperic Administration Photo Library. [Online]. Avaiable: http://www.photolib.noaa.gov/.

ACKNOWLEDGMENTS

“Trust in the Lord with all your heart, and lean not on your own understanding; in all your ways acknowledge Him, and He shall direct your paths.” Proverbs 3:5–6

Thank you, Lord Jesus, for the influence of the Holy Spirit and for blessing me with the support of my wife, Megan, and help and direction of my advisors, Dr. Mathias Kölsch and Dr. Douglas Fouts.

“You are worthy, O Lord, to receive glory and honor and power; for You created all things, and by Your will they exist and were created.” Revelation 4:11

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. COMPUTER VISION SHIP DETECTION FOR NAVAL FORCES

The U.S. Navy continues to rely on computers to perform tasks that were previously done by operators. Many benefits are gained by having a computer perform a task rather than an operator, such as enabling operators to be more effective at their missions. A computer can do visual ship detection, though there are many challenges to overcome. The focus of this research is on evaluating object detection methods as a viable replacement or aid for operators to continuously search for surface contacts with a photonics mast visual sensor. Many object detection methods are considered, and four approaches were chosen for evaluation as possible visual ship detectors for naval forces.

B. BENEFITS OF AUTOMATIC VISUAL SHIP DETECTION

When operating in a nautical environment there are many choices of sensors for detecting and tracking other vessels. In many cases, the searcher may want to remain undetected themselves, especially if the operational priorities for detection are passive intelligence, surveillance and recognizance (ISR). Unmanned vehicles, such as unmanned aerial vehicles, unmanned surface vehicles or unmanned undersea vehicles, could be conducting ISR and would benefit from computer vision ship detection. Submarines currently conduct ISR missions with priority to use passive detection methods and sensors only. One of the most useful and accurate sensors is an optical sensor. Most optical sensors, including a periscope or photonics mast, require continual observation by an operator to detect, classify and track any contacts acquired from these sensors. Other sensors such as sound navigation and ranging (SONAR) and radio detection and ranging (RADAR) systems have had faster integration of automatic detection, classification and tracking systems. The same automatic detection could be available for an optical sensor through computer vision with the rapid advancements and the creation of many precise and robust object detections methods.

Not all contacts can be detected by passive RADAR and SONAR systems; these contacts may require an optical sensor for detection and tracking. Passive RADAR will only pick up contacts that are emitting or reflecting a RADAR signals. Contacts can also be too quiet to detect with SONAR or the acoustic environment can limit detection. An optical sensor can detect many of these contacts that are undetectable by passive SONAR and RADAR. Visual detection of a contact has additional benefits. For example, a visually acquired contact can be easily classified. After classifying a contact visually, the range can be calculated quickly and with every visual observation. Bearing accuracy is another benefit of visual detection, along with providing the direction the contact is facing and, subsequently, the direction of movement.

Visual detection of a contact can give course, bearing and range in a single observation and speed with follow-on observations. Managing multiple visual contacts at once can be difficult for operators. An operator can only make an observation on a single contact at a time. An observation records the contact bearing and possibly records an image. Range can be determined from the height of the contact. Course can be determined by the direction the bow is facing and calculated from the observer's point of view. Speed can be calculated from follow-on observations using the change in bearing and range. Detecting and tracking ships visually can take two people, one to operate the sensor the other to record the data. This process can be rather quick for a proficient team or can be very slow for people in training. The same process must be repeated frequently for every contact to continuously update the contact picture.

The quality of information gathered from an optical sensor is dependent on the operators. The probability of detection of a distant vessel is also dependent on the operators. An operator's visual acuity contributes to the probability of detection, along with fatigue, strain, training and proficiency. There are other factors such as weather, sea state and the position of sun that can change the probability of detecting a ship. Detecting ships with computer vision algorithms would reduce or eliminate many of the issues that operators face. A computer would have a known visual acuity, probability of detection and scan rate. These parameters would not change with fatigue as they do with an operator. When a computer can detect every ship that is seen visually with a 100 percent probability of detection and an instantaneous rate of scan, it would be obvious to have a computer take over this function. Currently, 100 percent detection probability is not realistic for a computer or an operator but is it time for a computer to be the primary visual ship detector?

C. CHALLENGES OF AUTOMATIC SHIP DETECTION

There are many challenges for an operator or for a computer when detecting ships in open seas and harbor environments. For open seas ship detection, distance between ship and sensor can provide challenges. As the distance increases, the size of the ship in the image decreases, only being blocked by the horizon. The horizon and great distance can cause other problems such as mirage, an optical distortion that may make the ship unrecognizable. Weather can also be challenging for open seas and in harbors. A challenge common in harbor ship detection is that buildings can closely resemble ship superstructures. For a computer to replace operators for ship detection, it should be able to overcome the challenges of open seas, harbor and costal ship detection. To take into account these challenges, training and testing images were selected to incorporate many of these challenges. The evaluation images were scaled down to approximate ships at great distances. Images that contain many buildings and no ships are used to challenge the ship detection methods for operations in coastal waters.

D. MODELING THE EVALUATION BASED ON SEARCH THEORY

To frame the evaluation around performing the task of visual search and detection with a computer, a search theory approach that has been developed for evaluating SONAR and RADAR systems was adapted. Multiple computer vision object detection algorithms were evaluated with the search theory framework. The framework provided useful information to compare the computer vision algorithms as operational ship detectors. To determine how well a computer performed the task of visual ship detection, the performance of recall, precision and time to process an image frame were measured. These parameters were used to calculate a probability of detection from a single glimpse δ by range and a detection probability per sensor sweep P_D by range. Finally, the search theory model combines all of the parameters into a single number defined as sweep width

ω to directly compare each detection method. The creation of the evaluation model allowed for still imagery to represent the challenges of operational ship detection without performing an actual operational test.

E. FOCUS OF THIS RESEARCH

The investigation of the ship detection was confined to the horizontal aspect, so the sensor detecting the ship was in the same horizontal plane as the ship being detected. The first concern was "Can a computer detect a ship with enough precision and detection probability to be a viable alternative to a human operator?" The selection of algorithms to test was based on published results of the detection probabilities for other objects since there was very limited work in the field of visual ship detection from the horizontal aspect. Much of the research in visual ship detection is from overhead and satellite point of view. The initial investigation for this thesis considered visual detection approaches used for satellites through collaboration with K. Rainey et al. [1]. The initial investigation found ship detection from a photonics mast to be a much different problem than ship detection from a satellite, with larger variances in size and aspects of ships. Realizing these differences led the focus of the research towards general object detection methods instead of following satellite ship detection methods.

All the detection methods evaluated were able to detect ships of multiple sizes, scales and aspects. Evaluating these methods based on recall and precision for ship detection was the first priority. The second priority was ensuring that the algorithms could detect ships fast enough to be a viable alternative or aid to an operator. Multicore, multithreaded, parallel computations of central processing units (CPUs) and parallel computation on graphical processors (GPUs) was used to optimize the detection methods for the evaluation hardware.

F. OBJECT DETECTION METHODS EVALUATED AS SHIP DETECTORS

The challenges found for visual ship detection led the investigation to consider three classes of object detection methods. A fourth approach was developed by sending the positive detection of one method to another method for additional evaluation. The first method evaluated was a bag of keypoints or bag of words (BOW) method introduced by Csurka et al. [2]. The BOW method was evaluated as a two-class detector, detecting either ships or non-ships. A benefit of the BOW approach was it can be extended to multi-class detection and can be trained to detect ships of different classes as done by General Dynamics [3]. The second object detection approach evaluated was a histogram of oriented gradients (HOG) method developed by Dalal and Triggs [4]. HOG has been very successful at tracking pedestrians and can be implemented and optimized for computation on a GPU, as shown by Lillywhite et al. [5]. The third method evaluated was a deformable parts model (DPM) based on work by D. Ramanan et al. [6]. DPM object detectors have been used for detecting many types of objects and are commonly used in pattern analysis, statistical modeling and computational learning (PASCAL2) visual object classes challenge (VOC) [7].

Many of the most successful object detection methods from PASCAL2 VOC have been hybrid detection models. The fourth ship detector evaluated was a hybrid model. The hybrid method (HYBRID) was constructed by sending the results of the HOG detector to the BOW classifier. The HYBRID detector was created to take advantage of the fast computation of the HOG detection method done on a GPU while having the option to extend it to multi-class detection provided by the BOW model. The HYBRID was able to reduce the false positive rate while maintaining or increasing recall at a cost of increasing computational time and complexity.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

The field of object detection in computer vision was very large, whereas the subset of computer vision for ship detection was very small. Research of ship detection in computer vision has been done for multiple purposes, from detecting ships in satellite imagery or for harbor surveillance, to detecting ships as a general object as part of the PASCAL2 VOC. Previously developed ship detectors and the selected object detection methods are analyzed in this chapter. The selected object detection methods are broken down into computational stages of preprocessing, feature detection, feature extraction and feature comparison. Feature comparison that utilizes machine learning for matching of features is also discussed.

A. PREVIOUS COMPUTER VISION SHIP DETECTION

The majority of research published on ship detection from the horizontal point of view is for the purpose of harbor surveillance. General Dynamics investigated an object detection method using scale-invariant feature transform (SIFT) [8] to identify ships in a harbor [3]. The National Ocean Technology Center in Tianjin, China investigated harbor surveillance in *Ship Tracking Using Background Subtraction and Inter-frame Correlation* [9]. S. Fefilatyev et al. at the University of South Florida conducted an investigation for maritime surveillance using an autonomous buoy with an automatic identification system (AIS) and an onboard camera and video processing computer [10,11,12].

Some of the most robust algorithms investigated for ship detection have been submitted to the PASCAL2 VOC [7]. In 2007, the VOC added boats as one of the 20 classes in the challenge. The VOC receives many competitors every year, and the challenge was based on detecting individual classes of objects in images containing multiple classes of objects. The evaluation approach of this research differs by only looking for ships in images that could contain ships. The algorithms that have had success at classifying and detecting boats for the VOC have been BOW, HOG and DPM methods. The best boat detector from 2012 VOC had an average precision (AP%) of

24.8 and was a hybrid BOW that used segmentation for localization [13]. The second best boat detector used HOG features and local binary patterns; the third best was a DPM algorithm that used multiple-kernel learning for the support vector machine (SVM) [13].

B. PREPROCESSING AND PRE-FILTERING

In many cases, when a detection method has more computational intensive stages, a fast preprocessing method was used to find regions of interest (ROI). Many preprocessing methods require a video stream or data from previous frames. A video stream allows for background subtraction of the sky and ocean, as done by the National Ocean Technology Center in *Ship Tracking with Background Subtraction* [9]. A preprocessing stage that performs horizon detections was also found to be beneficial for ship detection from a buoy by Fefilatyev et al. [12]. These preprocessing stages are beneficial in their applications; although, it was found difficult to apply to a submarine photonics mast that would be continually moving and rotating. The *Ship Tracking with Background Subtraction* algorithm was highly dependent on background subtraction, using frame differences to find motion in the image as a possible ship for processing in future stages. Background subtraction was accomplished by removing mean image color and intensity of the background over frame averaging from subsequent frames [9]. A moving sensor may cause the appearance that every pixel is moving, providing difficulties for background subtraction. Background subtraction could not be applied using a similar method since this evaluation was conducted on still images.

The ship detection method that used horizon detection was limited by the dependence of this preprocessing stage. In their ship detection from a buoy-mounted camera, Fefilatyev et al. [12] selected ROI that extend above the horizon as possible ships. Any frame where the horizon was not detected was rejected for further processing. The horizon localization approach was not intended for use on a horizon that might be a coastline, which may contain ROI that are buildings and not ships. The horizon was found by segmenting the image into sky and ocean. Many of the evaluation images contained ocean, land and sky and Fefilatyev et al. did not account for land in their horizon detector [14]. None of the ship detection methods evaluated required these preprocessing stages and none were applied. The only preprocessing that any of the methods required was digital zoom or pre-scaling. Many of the features of the ship detectors required more pixels to create the feature than were in the smaller scale images. Digital zoom was accomplished by linear interpolation to double the size of the images and improved results in many cases. The two times pre-scaling increased the computation time for future stages by providing four times the pixels for finding features.

C. FEATURE AND KEYPOINT LOCATIONS

There are many different processes for locating interest or keypoints in images and many ways to describe them or extract feature descriptors from these keypoints. Some features come from other image processing techniques such as splicing together images, while others have been developed purely for use in detection and classification of objects. The general BOW detection method provides a means to independently evaluate many keypoint selection methods, feature descriptor methods and descriptor matching methods. All of the keypoint selecting methods have their advantages and disadvantages, such as some may hold more information and get better matching result while others may be faster to compute and match. Many keypoint selection algorithms were considered in the evaluation of the BOW method as a ship detector.

Most of the processes to extract keypoints transform the image from color intensity into gradients of intensity such as the Harris corner detector (HARRIS) [15]. Features from accelerated segment test (FAST) built upon HARRIS, with additional thresholding and non-maximum suppression, to quickly select repeatable corners, and FAST was more adept to invariants than its predecessors [16]. SIFT extended this gradient approach by selecting keypoints through a difference of Gaussian filter (DoG) and a Hessian matrix [8]. Speeded up robust features (SURF) was created to address the high dimensionality of the SIFT descriptor to reduce the description and matching time [17]. SURF introduced a box filter approximation to calculate the DoG for the Hessian matrix keypoint selector. Center surround extrema (CenSurE) as introduced in [18] and subsequently described as (STAR) in [19] was developed after SIFT and SURF but uses the HARRIS edge filter to reject features. STAR also uses a Laplacian approximation by calculating difference of boxes where one box resides entirely in the other box. STAR is more similar to SURF than SIFT. Good features to track (GFTT) is one of the earliest feature selection methods predating HARRIS and used Eigenvalues to measure "texturedness" of image intensity to select keypoints [20]. Another method is maximally stable extremal region (MSER) where the features are created by thresholding the grayscale intensity of the image and selecting regional outliers from these thresholds [21]. All of these methods for locating keypoints are available in the OpenCV library [19] and have been evaluated with the BOW ship detector in Chapter III.

Recently, there has been an increase in the use of features that are described as binary features. They are classified as binary features because they use a binary representation method instead of a floating-point representation. Binary should not be confused with integer, since the binary representation uses every bit as part of the descriptor, where an integer representation is a sequential number representation. The typical distance of measure between binary features is the Hamming distance. Three of these binary feature descriptor methods are binary robust independent elementary features (BRIEF) [22], oriented fast and rotated BRIEF (ORB) [23] and binary robust invariant scalable keypoints (BRISK) [24]. Both ORB and BRISK were evaluated as feature selection methods for the BOW ship detector; ORB is the extended version of BRIEF; their associated feature descriptor and matching methods could not be evaluated in the BOW model. The binary representation was not compatible with the BOW descriptor and matching methods since the implementation required a floating-point descriptor. Even though the descriptor had to be floating-point, the method that selected the keypoints to create the descriptors did not require a floating-point selection method. Fewer restrictions on the keypoint selector allowed the comparison of many more keypoint selecting methods than descriptor and matching methods.

D. DESCRIPTOR CREATION

Once keypoint locations have been selected, the features at these locations in the image are described in a vocabulary or dictionary by a descriptor. The descriptors are matched or compared in later stages through a machine-learning algorithm. Two feature descriptors, SIFT and SURF, were chosen to evaluate through the BOW method with two additional variants, modified by the opponent color space as described by E. Koen et al. [25]. The opponent color space allowed comparisons of OpponentSIFT and OpponentSURF descriptors that extend the SIFT and SURF descriptor over the individual Red, Green, Blue (RGB) color spaces. OpponentSIFT and OpponentSURF have been shown to have better results for category recognition by K. van de Sande et al [25]. SIFT and SURF are the most prevalently used floating-point descriptors. The implementation of binary descriptors for image matching has been shown to be an efficient alternative to SIFT or SURF [23] but not yet for object detection. The BOW model is not inherently compatible with binary descriptors since it is based on machine learning comparison through a SVM that relies on floating point descriptors.

The BOW implementation locates keypoints to be the selected feature locations. The descriptors are calculated at these keypoints instead of at every location in the images. The HOG and DPM detectors that were selected skip both the preprocessing stage and the keypoint selection stage and instead create descriptors of the entire image and at multiple scales. Both these detectors transform the image into a grid of HOGs. The image is transformed first into the vertical and horizontal gradient such as in many of the keypoint location methods. Then the entire image is divided into small regions and described as HOGs. The HOG detector in this research, like the original for pedestrian detection [4], scans the detector descriptor over the image regional descriptors looking for possible matches. HOG and DPM are also scale invariant and can detect objects that are larger than the detector descriptor by scaling the image down and running the detection again. The only way to find objects smaller than the original detector is to scale the image up before running the detector, which is one of the reasons for pre-scaling as a preprocessing stage. The DPM detector was built on the OpenCV DPM library from H. Bristow [26], implementing the DPM detection method described by D. Ramanan et al. [27]. The DPM detector HOG descriptors are constructed for individual parts. DPM also adds an additional constraint on the locations of these parts relative to each. For BOW, HOG and DPM to detect a ship, the descriptors from the detector and the image must meet matching criteria.

E. MACHINE LEARNING AND DESCRIPTOR MATCHING

The last stage for detecting a ship was the matching of the descriptors where a decision was made to declare a positive detection. The complexity of matching descriptors from the detector and the image increases with the size and number of descriptors. Machine learning was used to reduce the number of comparisons. Some approaches still compare every descriptor from a training database to every descriptor in the evaluation imagery such as used in by the General Dynamics study to identify a ship down to the exact ship [3]. Machine learning can reduce the descriptors from all of the training images down to a single descriptor for comparison with the evaluation images. The two most widely used machine-learning approaches for object detection are Naïve Bayes and SVM. Training in this research was conducted with SVMs, which have been found to "substantially outperform Naïve Bayes" [28], when training on large data sets with widely varying data; the ship images used here varied greatly.

A matcher compares descriptors by measuring the distance between descriptors. There are many methods of measuring distances between descriptors. For HOG and DPM, this distance was the geometric distance in the dimensional space created by the feature descriptor and was done by a linear SVM. DPM extends the linear SVM to a SVM with latent variables for the distances and relationship among the parts [27]. Three matchers for the BOW were tested. Two matched with brute force algorithms, measuring the distance between every descriptor of the detector and the evaluation image. The difference in the brute force methods was one used Manhattan distance (L1) and the other used Euclidean distance (L2). The third matcher also used L2 and was the fast library approximate of nearest neighbors (FLANN). FLANN was developed to reduce the number of comparisons improving computational speed [29].

Even though the brute force methods compare every descriptor, the number of descriptors can be specified. BOW method clusters the training descriptors into the desired number of words to describe the object being trained on, in this case ships. A point that best represents the cluster center was then selected as the word for all those descriptors; the collection of all of the words was called the vocabulary as described in an initial Xerox research report [2]. BOW also clusters the evaluation image descriptors into a vocabulary of the same size. A match was called if the calculated distances are below a threshold. The threshold was initially set by the SVM as the distance from the detector descriptors to the hyper plane that best separates the training positive descriptors from the training negative descriptors. This threshold was adjusted to create the receiver operator characteristic (ROC) curves in the evaluation.

F. EVALUATION CRITERIA

The evaluation method takes into account the application for using the computer vision algorithms as a visual ship detector sensor on a submarine photonics mast. In search theory, there are evaluation methods for comparing sensors from SONAR and RADAR to searching with the human eye. The evaluation model was an adaptation of a method for comparing sensors in the field of SONAR and RADAR that are discussed and summarized in *Search and Detection* [30] and in *Naval Operations Analysis* [31]. SONAR and RADAR evaluation methods define the probability of detection for a discrete glimpse based on the contact known signal strength, signal-to-noise ratio and signal excess. This evaluation differs by defining detection probabilities based on experimentally obtaining recall from the evaluation images. The evaluation method is dependent on the time between glimpses of the target. This time was determined by the computational time of the detection methods and hardware limitation described in III.B Evaluation Method and Criteria. The dependencies on computational time led to the consideration of optimization for multicore CPUs and GPUs.

G. PARALLEL PROCESSING FOR COMPUTATION SPEED GAINS

Most computer vision image processes are inherently parallel. The evaluated detectors were built upon the OpenCV library [19]. The matrix operations and image processing function of the OpenCV library are optimized for multicore CPUs by using OpenMP [32] or Intel Thread Building Blocks (TBB) [33]. More recently, algorithms of the OpenCV library have been ported to GPUs through both OpenCL [34] and CUDA [35]. A seven times speed up for a pedestrian HOG detector using the OpenCV GPU methods versus the multicore CPU methods has been demonstrated by A. Baksheev et al. [36]. This HOG detector is very similar to the implementation constructed here, though comparable results were not obtained. The published seven times speed up did not take into account the memory transfer time from the main memory to the GPU memory, which was considered in the evaluation done here. This research also considered much larger imagery at 1920 by 1080 pixels, compared to the 640 by 480 pixels.

III. METHODS

A. TRAINING AND SELECTION AS A DESIGN OF EXPERIMENTS

Training was conducted for all detectors as a design of experiments; the first priority was to maximizing each object detection method's probability of detecting ships. Each detector had different parameters that were adjusted during the training and are discussed in the beginning of that method's section. Careful selection was used for both the evaluation and training data, ensuring the evaluation set was represented well by the training data, as recommend by Zhu et al. [37]. Also, the training data was accurately labeled as recommended by J. Ponce et al. [38]. Each training session took on the order of minutes to days and to train this large number of detectors, the Naval Postgraduate School high performance computer "Hamming" was utilized. The ship detectors were narrowed down based on recall of ships to the best few for each method. These top detectors for each method were then evaluated for computational speed on specific hardware. Finally, the best ship detector from each method was compared in Chapter IV based on the evaluation model.

B. EVALUATION METHOD AND CRITERIA

The evaluation method was based on the concept of lateral range curves and sweep width ω, from the field of search theory, discussed in the textbooks *Naval Operations Analysis* [31] and *Search and Detection* [30]. The ω was considered the expected detection range of the sensor when allocating sensors to search for a target [31]. This evaluation model allowed the comparison of the different ship detection algorithms to each other in a manner relevant for operational ship detection from a photonics mast. In constructing the operational model, the evaluation images were meticulously labeled to obtain the pixel height of all ships. The Kodak KAI-2093 was assumed throughout the model to be the image equitation sensor. The sensor was assumed to have taken all of the evaluation imagery to model the size of ships in the images to a corresponding range. The probability of detecting a ship in a single observation, or "glimpse," was determined over multiple ranges by experimentally obtaining recall probabilities over multiple scales of images. The evaluation model also took into account the computational time for each algorithm to continuously scan 360-degrees.

The evaluation model also took into consideration that there would be physical hardware driving the rotation of the sensor. The visual sensor could cause optical distortion by rotating too quickly during the exposure time. To prevent this type of distortion, limits were set on the rotation rate of the sensor. The frame rate was also limited to 30 frames per second (fps). The time to detect a ship in each image frame was calculated using identical hardware and conditions for all methods. The 360-degree scans were required to be continuous so no bearing was ever missed. Additionally, overlap was required to ensure a ship could not pass between frames without detection. The time the sensor took to make a complete sweep was limited by either the rotating hardware or the processing time of the detection method. The time to process an image frame and the time to complete a sensor sweep were used when calculating the detection probability for a given range of a ship. The evaluation model limits the range to be the closest point of approach (CPA). The lateral range curves were then constructed by graphing the probabilities of detection for multiple detection ranges. The ω was calculated as the area under the lateral range curve as defined in *Naval Operation Analysis*. The ω is considered the expected detection range of a sensor when allocating sensors to search for a target [31]. The ω provided a single number to compare the detection methods.

1. The Evaluation Images

The evaluation set consisted of 405 positive images that contain a ship and 100 negative images that did not contain a ship. The positive images were selected to contain a wide variety of ship types from small speedboats to aircraft carriers and submarines. The images were also selected on a basis of being from a sea level vantage point. This point of view best represents the perspective of a submarine photonics mast. The negative evaluation image set contained scenes of ocean, sky, coastline with and without buildings and some ocean pictures that contained creatures, such as whales and birds. The image set was selected to best represent the environment in which a submarine would be operating. Many of the positive and all of the negative images came from the

National Oceanic and Atmospheric Association (NOAA) [39]. Images where also obtained from Jane's Fighting Ships [40] and http://www.shipphotos.co.uk/ [41]. None of the images were taken with the 1920 by 1080 sensor used in the evaluation model. The positive evaluation images had a mean size of 645 by 451 pixels; a sample of six images is shown in Figure 1. The negative images had a mean size of 1564 by 1078 pixels, and a sample of these images is shown in Figure 2. The size of the images was not as important to the evaluation model as the size of the ships in the images. The heights of the ships in pixels are the inputs to calculate observational ranges.

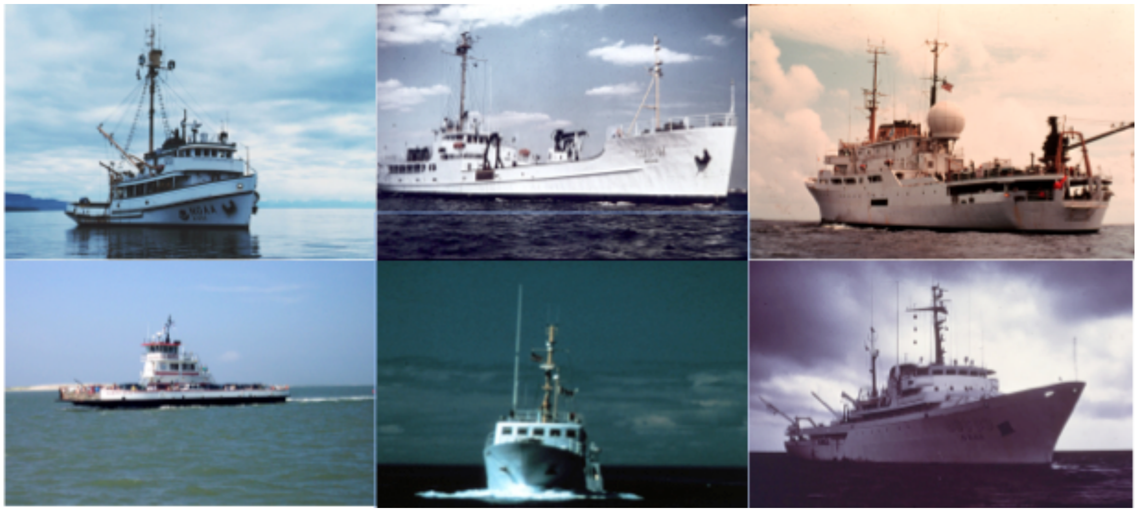

Figure 1. Samples of the positive evaluation images (from [39]).

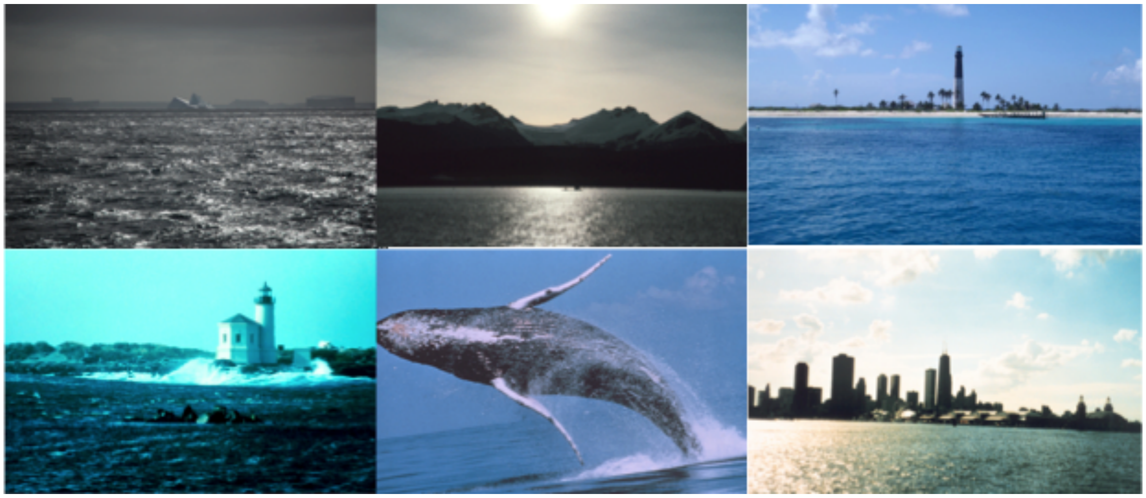

Figure 2. Samples of the negative evaluation images (from [39]).

2. Calculating Range by Pixels

The actual sensor that recorded each image was unknown. Knowing the hardware that acquired the image of the ship is critical for determining the range at which the sensor was from the ship. The KAI-2093 was assumed to have produced all of the images with four possible lens configurations. The KAI-2093 pixel array was 1920 x 1080 square pixels of 7.4 μm per side, resulting in a sensor height of 8 mm [42]. The four lens types simulate low, medium, high and super high power zoom. The focal lengths f_l considered are 7, 46, 140 and 280 mm, giving a field-of-view (FOV) of 36, 16, 6, and 2.4 degrees, respectively. The FOVs are labeled as wide, medium, narrow and ultra-narrow, or WFOV, MFOV, NFOV and UNFOV, respectively.

The pinhole camera equations in *Computer Vision: A Modern Approach* [43] provided two equal ratios. The first was the vertical resolution in pixels h_{sp}, 1080 for the KAI-2093, and f_l over the sensor height in meters h_{sm}, 8 mm for the KAI-2093. The second was the range R from the pinhole to the ship and the height of the ship in pixels m_p over physical height of the ship in meters m_m. Range of observation was calculated by rearranging these ratios as

$$R = \frac{f_l\, m_m\, h_{sp}}{h_{sm}\, m_p}. \tag{1}$$

Using one of the four lenses with a different f_l varies the range of observation. The drawback of a longer range was a narrower FOV. The m_p was measured on a per image basis since all positive evaluation images were labeled with LabelMe [44]. The m_m is the independent variable for range in equation (1) to calculate the range for ships of different physical heights.

The model also takes into account the distance to the horizon and how a ship disappears over the horizon as the range increases. The maximum range in meters for two contacts in visual sight is [45]:

$$R_{\max} = 3570\left(\sqrt{h_s} + \sqrt{m_m}\right). \tag{2}$$

The height of the ship below the horizon m_{bh} and the actual range R to the contact were calculated by rearranging equation (2):

$$m_{bh} = \left(\frac{R}{3570} + \sqrt{h_s} \right)^2 . \tag{3}$$

The height of the ship visible above the horizon m_{ah} is then

$$m_{ah} = m_m - m_{bh} . \tag{4}$$

A diagram displaying the over-the-horizon range variables is illustrated in Figure 3. Given an observer's sensors h_s, the height of the photonics mast of two meters yields a distance to the horizon of 5.05 kilometers. Two equations can then be developed for m_p. The first equation is for ranges less than the distance to the horizon:

$$m_p = \frac{f_l m_m h_{sp}}{R h_{sm}} . \tag{5}$$

The second equation is for contacts at ranges greater than the distance to the horizon:

$$m_p = \frac{f_l \, h_{sp} \left(m_m - \left(R/3570 - \sqrt{h_s} \right)^2 \right)}{R} . \tag{6}$$

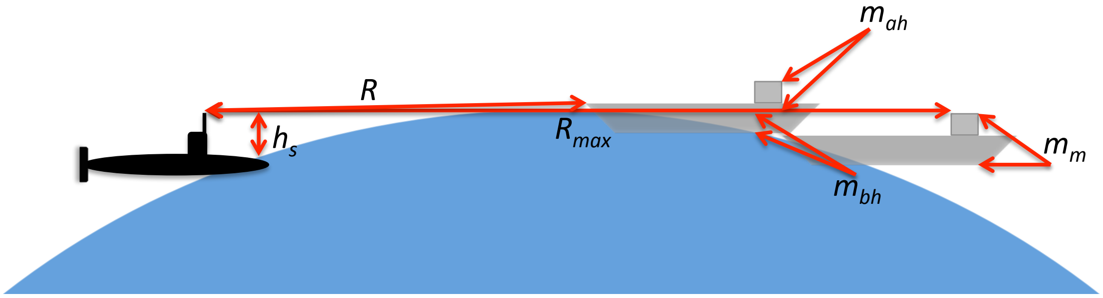

Figure 3. Over the horizon, line-of-sight range diagram.

Illustrated in Figure 4 and Figure 5 are the pixel heights for a 30 meter (98.4 feet) and 10 meter (32.8 feet) mast head height ship with respect to its range. These graphs were constructed with equation (5) for ranges less than the distance to the horizon and equation (6) for ranges greater than the distance to the horizon. The graphs are plotted for ranges until the vessel was completely hidden by the horizon. The range at which half of the ship is hidden over the horizon was also labeled. Evaluations were not conducted on images in which half of the ship or more was over the horizon, and the evaluation assumed that the detection probability was zero beyond this point for all detectors.

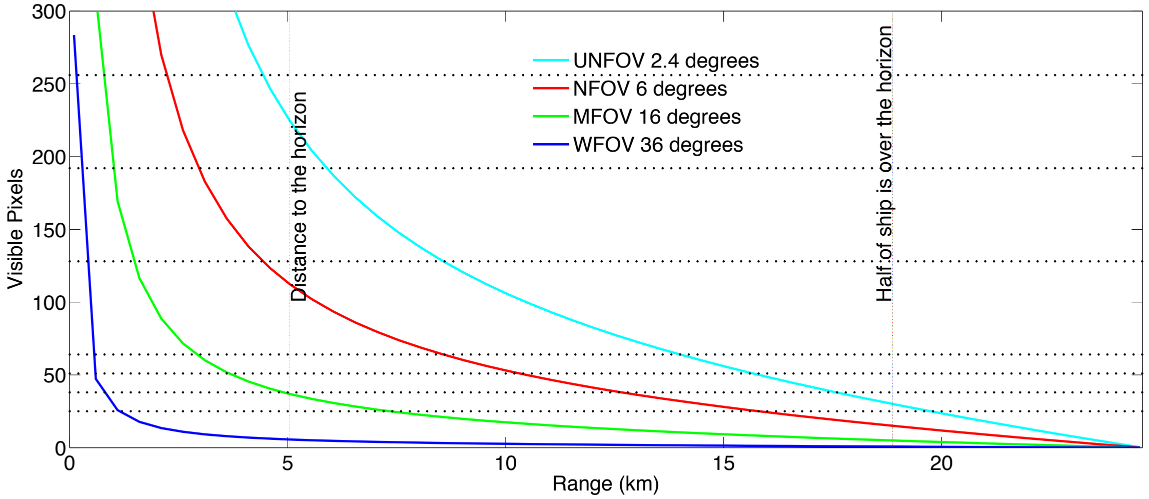

Figure 4. Graph of the visible height in pixels of a ship 30-meter MHH ship by range.

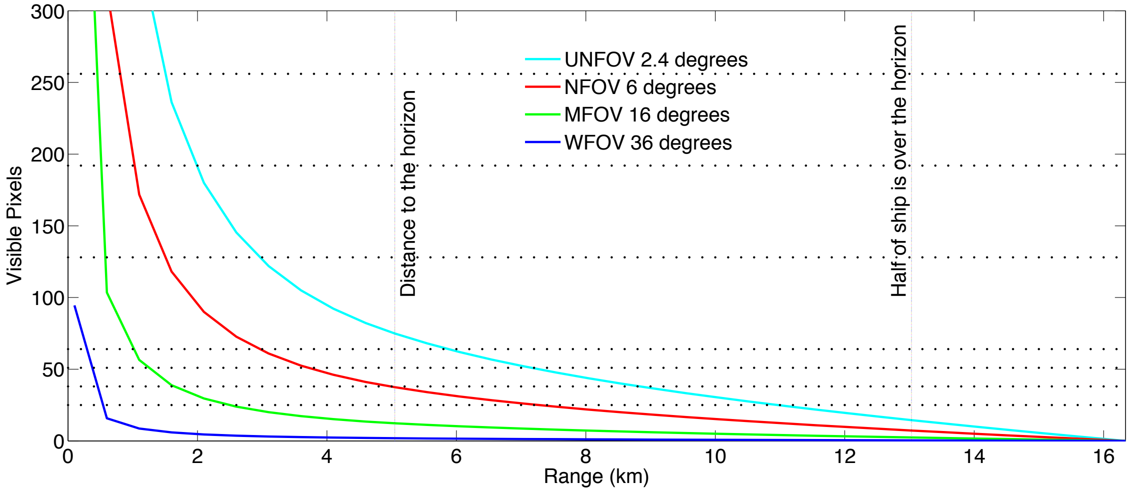

Figure 5. Graph of the visible height in pixels of a ship 10-meter MHH ship by range.

The dashed lines across the graphs in Figure 4 and Figure 5 at 256 pixels correspond to the mean m_p of the positive evaluation images. The m_p was calculated from the top of the highest mast to the lowest point on the waterline. The standard deviation of m_p was 85.4 pixels. The positive and negative evaluation images were scaled down to 75, 50, 25, 20, 15 and 10 percent of the original image height and width, preserving the aspect ratio. The other dashed lines in Figure 4 and Figure 5 correspond to the mean m_p values of the scaled down images with values of 192, 128, 64, 51, 38 and 25 pixels. The scaled down images simulate ships at greater distances. No additional noise was introduced to simulate any other type of optical distortion, such as haze or mirage. However, some of the positive evaluation images were already blurry from optical distortion. This simulation of ship detection at multiple ranges allowed for experimentally obtaining the probability of detection P_D at these ranges.

3. Experimentally Obtaining Detection Probability by Range

The evaluation model defines two detection probabilities. The first was the probability of detecting a ship from a single glimpse, the glimpse probability δ. The second was the probability of detecting a ship through multiple glimpses in a single sweep of the sensor P_D. The P_D was used as an upper bound for methods that can achieve multiple glimpses in a single sensor sweep. The δ was obtained for each range where the dotted lines cross the solid lines in Figure 5. These intersections correspond to the mean ship height for the scaled evaluation sets. The δ for the ranges of intersection were calculated using the Pythagorean theorem and were the point on the corresponding ROC curve that was closest to a true positive detection percentage (recall) of one and false positive detection percentage (1-precision) of zero. Two other methods of obtaining δ from an ROC curve were considered. The first was using the area under the recall and precision ROC curve, which is similar to how AP% is calculated for PASCAL2 VOC results [13]. Area under the curve was not implemented since all ROC curves that have an area under the curve of greater than one half and are concave down would have a higher δ using area under the curve than if calculated with the Pythagorean theorem. The last method considered was designating a fixed false positive rate. A fixed false positive rate is less comprehensive to the evaluation, and the false positive rates are compared in the Chapter IV.

For the cases in which only a single glimpse of the contact was obtained, P_D was equal to δ. In cases where multiple glimpses of the target was obtained in a single 360-

degree sensor sweep, the P_D should be increased. However, assuming statistically independent glimpses is considered unrealistic, making

$$P_D = 1 - \prod_{i=1}^{n} (1-\delta) \tag{7}$$

for n independent glimpses unrealistic [31]. A statistical independence model is unrealistic because if the detection method did not detect the ship upon the first glimpse, it cannot be guaranteed that the method would have the same probability of detecting the ship in the next frame, when conditions are similar [31]. Taking multiple shots with a camera in very rapid secession could yield some images that are clearer than others, changing the probability of detection for some images. To allow for an increase in probability for multiple glimpses that was lower than statistical independence, the δ was weighted by the number of glimpses in equation (7) by raising δ to the power of the number of glimpses n in

$$P_D = 1 - \prod_{i=1}^{n} \left(1-\delta^n\right). \tag{8}$$

The weighted glimpses probability method and the statistical independent methods are illustrated in Figure 6 with $\delta = 0.7$. As can be seen from the graphs, the weighted method approaches P_D = 100 percent more slowly than the statistical independence method. The upper bound of detection probability for methods that could achieve multiple glimpses in a single sensor sweep is given by equation (8).

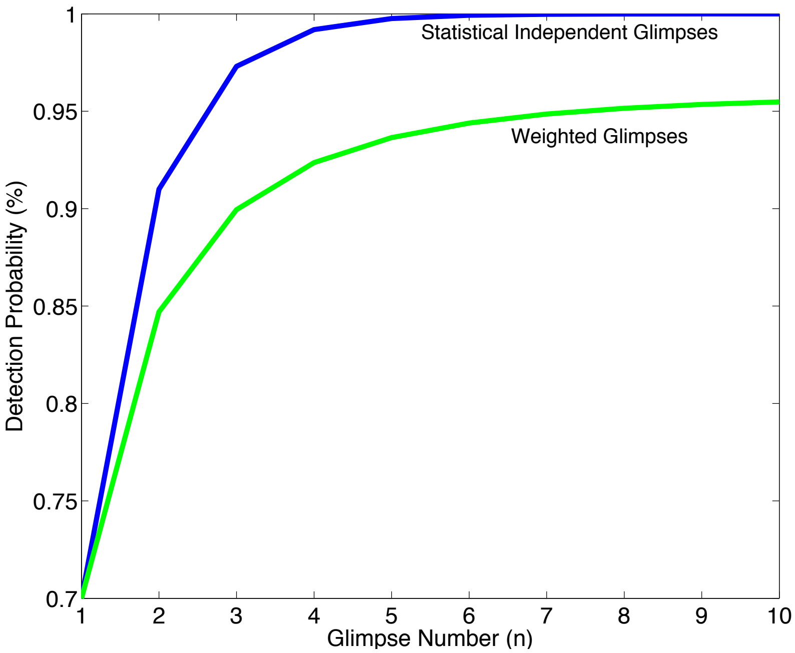

Figure 6. Example graph of the P_D calculated with δ of 0.7 for both statistically independent and weighted glimpses.

4. Calculating Time Per Frame and Sweep Time

The time to process one frame t_F was obtained experimentally. The time it takes for the detector to sweep 360-degrees, sweep time t_s, was limited by t_F or the hardware limits of the equipment. The KAI-2093 and the lenses under consideration do not have a 360-degree FOV; thus, a ship could be partially in one frame and partially in the next and not be detected. A ship could also travel opposite to the sweep direction and pass into a sector that was just scanned from the next sector to be scanned during the t_F. To minimize the probability of missing a ship entirely, when t_s was limited by t_F, the sensor was assumed to scan at the maximum rate to grab frames with a 10 percent overlap. Ship detector methods that process frames faster had their t_s limited by the assumed hardware, with t_s of 30 seconds for WFOV and MFOV and 60 seconds for NFOV and UNFOV. Limiting the t_s allowed faster detection methods to have an overlap percentage greater than 10 percent. When the overlap was large enough to have multiple glimpses of every bearing during a single sweep, multiple glimpses of contacts were considered for upper bounding P_D. The rotation rate or t_s limits were imposed to ensure adequate exposure time and were provided as hardware limitation of the sensor. The assumed minimum hardware limits of t_s were 30 seconds for WFOV and MFOV and 60 seconds for NFOV and UNFOV.

For t_s greater than the hardware limits, t_s was calculated as

$$t_s = t_F \times \frac{360 \times 1.1}{FOV} \quad . \tag{9}$$

Obtaining t_F was done by evaluating the detectors on an image set that consists of two 1920 by 1080 pixel images to match the KAI-2093. One image was positive and the other negative, so all detectors could produce 100 percent recall and 100 percent precision on these two images. This was important for the HOG, DPM and HYBRID methods since the computation time increased for the number of positive detections, whereas the BOW detection time did not change for positive detections. To minimize the impact that measurement time had on calculating timing and to include the memory transfer in the timing calculations, 100 copies of each of these images were created to comprise the 200 images for the t_F calculation test. The t_F was then averaged over five runs of the timing set to reduce effects of other processes that may have stolen processing time. All timing data does not contain the time to initialize the detectors and was gathered on the same computer under the same conditions.

The computer used to measure timing had an Intel® Core™ i7-3770K processor with four 64-bit cores and eight physical threads. The processor had three levels of cache: each core had a 64 kB and 256 kB level one and two cache, and all four cores share an 8 MB level three cache. The main memory was 16 GB of 1,600 MHz DDR3 RAM. The operating system was 64-bit and was located with the swap space on a SATA III SSD via a SATA III connection. All of the images were located on a separate SATA III SSD via a SATA II connection. The machine also had two graphics cards a NVIDIA GeForce® GTX 650 with 1 GB of DDR5 RAM, which was connected via a PCI Express 3.0 x 16 and was used to drive the monitor. The second card was a NVIDIA Quadro 2000 with 1 GB of DDR3 RAM connected via a PCI Express 3.0 x 4 and performed all GPU computations. The accuracy of the calculated recall and precision was verified for all methods in which the t_F and t_s were determined on this machine since all of the experimentally obtained ROC data was gathered on a cluster computer with a different operating system and libraries.

5. Constructing Lateral Range Curves and Calculating Sweep Width

The t_s was important for determining how far a ship traveled during the sweep time between possible observations. Lateral range curves are specific for a given contact. For this evaluation the lateral range curves are displayed for a 10-meter MHH contact traveling at a relative velocity of 20 knots (10.3 $^\mathrm{m}/_\mathrm{s}$). Velocity was always considered relative to the searcher. As in *Naval Operations Analysis* [31], the lateral range was calculated as the CPA of a passing contact. Visualization of how the glimpse probability δ for detecting a ship at range $\mathrm{r_m}$ becomes the probability of detecting a contact at lateral range x is illustrated in Figure 7. In order to guarantee at least one glimpse of the contact, one and one half complete sweeps of the sensor was required to be completed for the calculation of x. The extra half a sweep accounted for cases where a target entered the detection radius just after the sensor passed, then traveled behind the sensor scan, exiting the detection radius before the sensor could glimpse the contact. From Figure 7 the distance a contact travels inside the detection radius is $2y_o$. The evaluation model required the sensor complete one and a half sweeps prior to the contact traveling $2y_o$ for a contact with relative velocity v. Half the distance traveled by the contact inside the detection radius is then

$$y_o = \frac{1.5 t_s v}{2}. \tag{10}$$

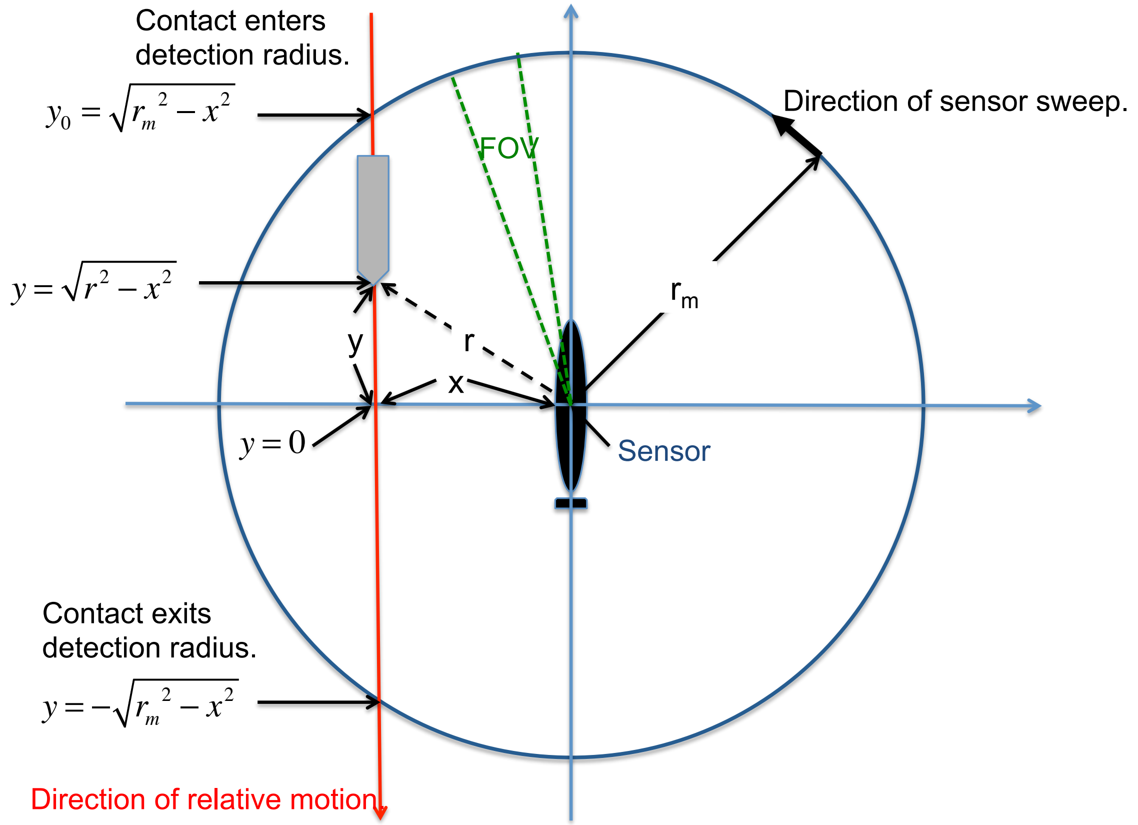
Contact enters detection radius.
$y_0 = \sqrt{r_m{}^2 - x^2}$
FOV
Direction of sensor sweep.
$y = \sqrt{r^2 - x^2}$
r
rm
y
x
$y = 0$
Sensor
Contact exits detection radius.
$y = -\sqrt{r_m{}^2 - x^2}$
Direction of relative motion

Figure 7. Lateral range calculation diagram (after [31]).

The lateral range x is calculated by rearranging the Figure 7 equation for x and substituting equation (10) for y_o. The lateral range is then

$$x = \sqrt{r_m{}^2 - \left(\frac{1.5t_s v}{2}\right)^2} . \tag{11}$$

There are two cases for calculating the P_D at x: the first for a single glimpse, the δ, the second is when t_s was limited by the hardware and multiple glimpses of a contact are possible. For the first case, there is at least 10 percent overlap of individual frames, and $P_D = \delta$. The second case is the upper bound of weighted P_D and was calculated from equation (8). The number of glimpses n was rounded down and calculated as

$$n = \frac{t_s FOV}{360 t_F} . \tag{12}$$

The maximum number of glimpses was limited by the 30 fps of the KAI-2093. Listed in Table 1 by FOV are the minimum sweep time, maximum number of glimpses, and minimum frame time required for two glimpses. The lateral range curves were constructed by plotting the δ by x. For the evaluation model there were seven sets of scaled images, each one providing one point on the lateral range curve. The expected detection range from each method was then derived as the area under the lateral range curve and is the sweep width ω. The ω was calculated by using a trapezoidal approximation of the area under the lateral range curves developed from δ and upper bounded by weighted P_D when possible.

Table 1. The minimum sweep time t_s, maximum glimpses n and minimum frame time t_F for two glimpses by FOV.

	$t_{\text{s minimum (seconds)}}$	$n_{maximum}$	$t_{\text{F for 2 glimpses (seconds)}}$
WFOV	30	90	1.5
MFOV	30	40	.67
NFOV	60	30	.5
UNFOV	60	12	.2

C. BAG OF WORDS

1. Design of Experiments for BOW

For the BOW model of detectors, over 2,000 detectors were trained and evaluated in the design of experiments. The parameters varied were the feature detector, descriptors, matchers, number of words and the number of training images. Initial tests for changing the number of training images did not greatly improve results beyond 100 positive and 100 negative images. Eventually, 105 positive and 105 negative images were used for training. A large improvement was observed from using labeled positive images cropped to a rectangle containing the top of the highest mast to the bottom of the hull and from bow to stern. Ten methods of extracting feature locations were evaluated. Four methods to describe these features location and three ways to match descriptors were examined. Initially, the number words were varied from 25 to 1,000. The BOW detectors performed better with less than 100, words and most BOW detectors performed the best with only 25 words. Additional evaluations with fewer than 25 words showed no increase in performance.

a. Locating Feature Keypoints

The nine feature detectors, as described in Chapter II.C, plus one additional feature detector were evaluated. The additional feature detector performed the worst, selected keypoints across the entire image at uniform intervals and is not shown in Figure 8. The nine other keypoint selection methods are shown in Figure 8 for the same image. This image displays how the algorithms can find many locations in the sky when cloudy and in the ocean from variances by waves. The number of keypoints selected by each algorithm varies from 471 with MSER to 11,263 with FAST. From the visualization of the keypoints, it is observed that SURF, SIFT, ORB and BRISK keypoints have location, magnitude and direction displayed. Magnitude is displayed by the size of the circle where direction is displayed by the line inside the circle. STAR and MSER have magnitude but no direction. FAST, GFTT and HARRIS have location displayed only, even though the keypoint retains an angle and a quality level. The HARRIS and GFTT keypoint selection methods are very similar in that they both execute the GFTT algorithm; the difference is that the HARRIS method enables the HARRIS corner detector to threshold keypoint locations for selection by removing lower quality points.

HARRIS did not seem to impact the GFTT method for reducing the number of keypoints selected since GFTT selected 1,000 and HARRIS also selected 1,000 keypoints. The default maximum number of keypoints for the GFTT and the HARRIS methods was 1,000. No adjustments were made for any of the methods to adjust the maximum number of allowed keypoints. In Figure 8, ORB does the best at selecting keypoints that are on the ship only. SURF appears to have selected the most keypoints in the clouds. SIFT, SURF and FAST all selected a large number of keypoints in the sea. STAR, MSER and BRISK selected fewer keypoints than the other methods and selected a smaller portion in the clouds but many in the sea. Even though the next stage requires the keypoints to be described by a floating-point feature of either SIFT or SURF, the locations where the features are described were selected by one of these nine methods. The OpenCV Library in which this BOW implementation was built upon contains more detailed descriptions of the implementations of the keypoint location, extraction and description methods [19].

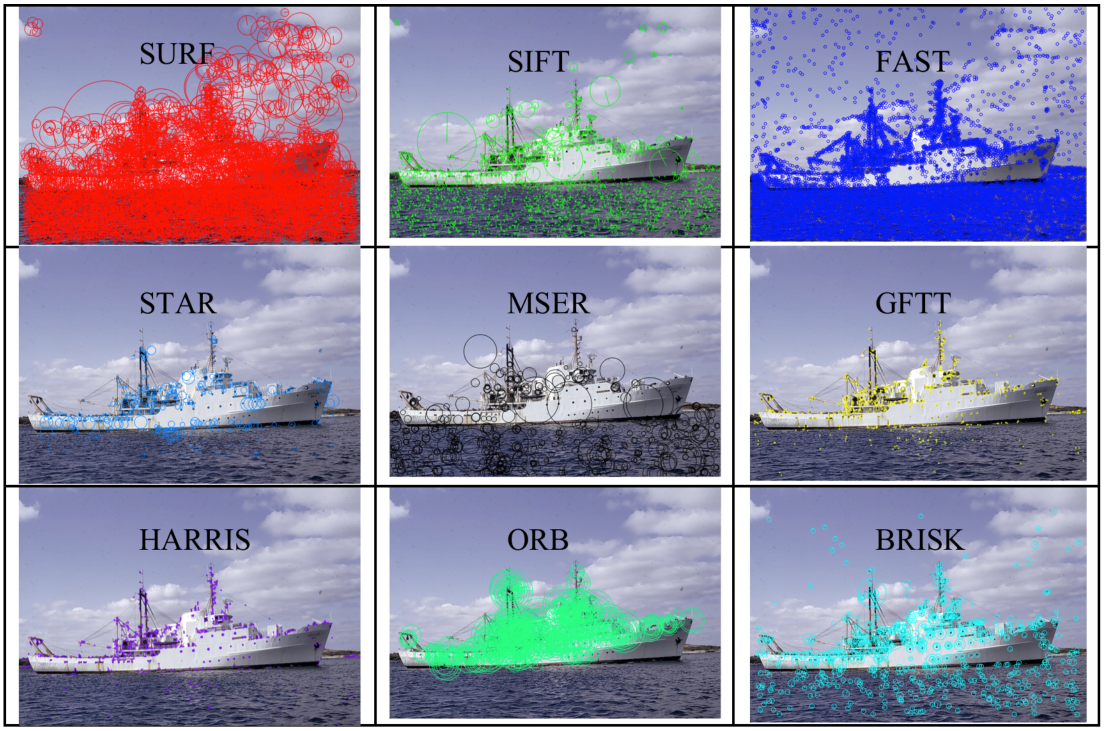

Figure 8. Visualization of nine of the different keypoint location methods evaluated through the BOW detector (after [39]).

b. Feature Extraction and Description

The descriptors created in the BOW model are the vocabularies created during training. Each vocabulary was limited in the number of words and was created to best represent the data in the positive training images by the SVM. The features can be considered the sentences that are created by the descriptors, words. Four descriptor methods were tested: SURF, SIFT, OpponentSURF and OpponentSIFT. The OpponentSURF and OpponentSIFT variants create a feature across the three separate color images, red, green and blue, and then combine them into one descriptor as described by K. E. A. van de Sande et al. [25]. The opponent methods required more processing time, and the results were not as promising as SURF and SIFT alone. SURF took the least processing time and produced the highest number of successful resulting detectors across all of the keypoint methods. The SIFT descriptor had many successful detectors as well and had the detector that had the closest recall and precision to 100 percent on the full size evaluation images. For the MSER and FAST keypoint selection methods, the SIFT descriptor greatly outperformed the SURF descriptor methods on recall and precision regardless of what matcher was used.

c. Matching the Descriptors

The three matchers were brute-force L1, brute-force L2 and FLANN. Detectors matching with L2 frequently outperformed detectors with the same keypoint selection and description methods, where L1 was the only difference. There were few occasions where L1 outperformed L2. There were also frequent occasions where FLANN and L2 produced identical results on the evaluation set for recall and precision. Interestingly, the method with fast in the name, FLANN, was typically the slowest method while L2 was typically the fastest. The matching methods had the smallest impact on the total time of computation. When two detectors returned the same results and only differed by the matcher, the fastest implementation was selected for further evaluation.

2. Selecting the Best BOW Detectors

In narrowing down the BOW methods to a few top detectors for comparison, every trained detector was evaluated on the full size images. From this initial evaluation, the top 400 BOW detectors were selected based on glimpse probability δ. The best two detectors of every feature selection method were also maintained, even if they were not in the top 400. Both the binary feature selection methods, ORB and BRISK, as well as DENSE, the uniform selection keypoint method, were not in the top 400. SURF, SIFT and MSER methods comprised the majority of the top 400 in this initial stage of reduction. In the next stage of selection, the top 400 detectors were tested on all of the scaled images with multiple threshold values. The top 100 detectors, including individual thresholding values, were selected from each scaled evaluation set; many of these detectors were the same detector. When a detector was in the top 100 of one scale, the probability was high that it would be near the top for multiple scales. Also, many of the detectors had multiple thresholding values that produced results in the top 100 of multiple scales. The SURF, SIFT and MSER keypoint selection methods again dominated the top 100 in almost every scale; besides these three methods, the best of each feature selection method was maintained for further evaluation. The detectors were narrowed down to 22 based on the detection probability alone.

3. Evaluation of Glimpse Probability for the Best BOW Detectors

The detectors were narrowed down to twelve detectors, maintaining the best detector for every feature method, except DENSE. Multiple detectors of SIFT and SURF were maintained, and the δ of these twelve are shown in Table 2. The δ was obtained from the ROC curves located in Appendix A, ROC Curves, Figure 26 through Figure 32. All feature location methods except the two binary methods had δ of greater than 90 percent for full size images; BRISK was very close at 89 percent. In Table 2, the highest δ for each scale is red. The best detector for each scale jumps around, yet the δ of many of the detectors are within a few percentage points. For the SIFT and SURF feature methods, no detector was clearly superior in all categories, each one having the highest δ for only one scale. The multiple detectors of SIFT and SURF keypoints provide a means to compare the effects of modifying the descriptor, matcher and number of words. The ORB detector, which selected all features on the ship in Figure 8, performed surprisingly poorly. ORB had the lowest δ for the two largest image sets. ORB and STAR keypoint methods require a larger number of pixels to calculate features. The requirement for a larger number of pixels prevented ORB and STAR to calculate features on the small images, subsequently, making their δ zero for small ships. Factoring in speed of computation was the next aspect of comparison for the evaluation model.

Table 2. Glimpse probabilities of the top 12 BOW detectors for evaluation image scales 100 percent to 10 percent.

Keypoint	Descriptor	Matcher	Words	100 (%)	75 (%)	50 (%)	25 (%)	20 (%)	15 (%)	10 (%)
SURF	SURF	L2	25	91	86	84	83	77	76	66
SURF	SURF	FLANN	50	92	90	84	77	82	74	70
SURF	SIFT	FLANN	25	97	90	93	78	66	78	66
SIFT	SURF	L1	25	98	88	88	79	70	66	56
SIFT	SURF	L2	25	97	92	84	77	75	63	60
FAST	SIFT	L2	50	92	89	83	71	74	72	71
STAR	SURF	FLANN	25	96	86	84	0	0	0	0
MSER	SIFT	L1	50	95	87	85	79	76	73	67
GFTT	SURF	L2	25	92	76	76	69	62	43	58
HARRIS	SURF	FLANN	100	95	83	72	76	56	72	62
ORB	SURF	FLANN	25	81	74	76	69	66	47	0
BRISK	SURF	L2	25	89	85	84	77	76	67	39

4. Evaluation of Frame Time and Number of Glimpses for the Best BOW Detectors

The descriptor extractor stage had the largest effect on the computation. After experimentally obtaining the t_F, the t_s were calculated from equation (9) and are listed in Table 3. When calculated t_s were less than the specified hardware-limited t_s, the hardware-limited t_s is listed instead. The disparity in computation time between SURF descriptor and SIFT descriptor is evident when comparing the SURF feature methods. The two SURF feature methods using SURF descriptors were over 10 times faster computationally than a SURF keypoint method that used SIFT descriptors. The SURF keypoint, SIFT descriptor, FLANN matcher detector had the highest δ for the 50 percent and 15 percent scaled images and nearly the highest in all other categories, yet it took over 17 minutes to perform a complete sweep in UNFOV. The MSER detector was nearly as slow computationally; part of the long computation time for MSER was due to the calculation of SIFT descriptors. FAST keypoint detectors also had the highest δ with SIFT descriptors and were nearly three times faster than the other methods that computed SIFT descriptors; however, they were still much slower than all SURF descriptor methods.

Table 3. Sweep time for the 12 best BOW detectors by FOV.

Keypoint	Descriptor	Matcher	Words	t_F (seconds)	*Sweep time* t_S (seconds)			
					WFOV	MFOV	NFOV	UNFOV
SURF	SURF	L2	25	0.554	30	30	60	91
SURF	SURF	FLANN	50	0.615	30	30	60	101
SURF	SIFT	FLANN	25	6.445	71	160	425	1,063
SIFT	SURF	L1	25	0.959	30	30	63	158
SIFT	SURF	L2	25	0.959	30	30	63	158
FAST	SIFT	L2	50	2.368	30	59	156	391
STAR	SURF	FLANN	25	0.145	30	30	60	60
MSER	SIFT	L1	50	6.147	68	152	406	1,014
GFTT	SURF	L2	25	0.169	30	30	60	60
HARRIS	SURF	FLANN	100	0.167	30	30	60	60
ORB	SURF	FLANN	25	0.101	30	30	60	60
BRISK	SURF	L2	25	0.086	30	30	60	60

Every detector with SURF descriptors had a t_F of less than one second. There was still an order of magnitude difference between the fastest detector, which used BRISK, and the slowest SURF descriptor detector; these differences had a large impact in the number of glimpses. The number of glimpses for the best BOW detectors are shown in Table 4. No detectors processed frames fast enough to reach the maximum number of glimpses imposed by the 30 fps of the KAI-2093. Many of the methods that had a higher δ had longer computation time and vice versa. Constructing the lateral range curve incorporates both glimpse probability and computation time in a manner relevant to the operational use of these methods on a submarine.

Table 4. Numbers of glimpses for the 12 best BOW detectors by FOV.

				t_F	n (number of glimpses)			
Keypoint	Descriptor	Matcher	Words	(seconds)	WFOV	MFOV	NFOV	UNFOV
SURF	SURF	L2	25	0.554	5	2	1	1
SURF	SURF	FLANN	50	0.615	4	2	1	1
SURF	SIFT	FLANN	25	6.445	1	1	1	1
SIFT	SURF	L1	25	0.959	3	1	1	1
SIFT	SURF	L2	25	0.959	3	1	1	1
FAST	SIFT	L2	50	2.368	1	1	1	1
STAR	SURF	FLANN	25	0.145	20	9	6	2
MSER	SIFT	L1	50	6.147	1	1	1	1
GFTT	SURF	L2	25	0.169	17	7	5	2
HARRIS	SURF	FLANN	100	0.167	17	7	5	2
ORB	SURF	FLANN	25	0.101	29	13	9	3
BRISK	SURF	L2	25	0.086	34	15	11	4

5. Lateral Range Curve and Sweep Width

The lateral range curves were constructed for the twelve best BOW detectors using δ for a 10 meter MHH contact with a relative speed of 20 knots, as described in Chapter III.B. The lateral range curves are presented in Appendix B, Lateral Range Curves, Figure 56 through Figure 59. The sweep width ω was calculated from these lateral range curves and is displayed in Table 5. The two SURF keypoint SURF descriptor detectors had the highest ω for all FOVs. The BRISK and HARRIS detectors have high ω for all FOVs even though they had some of the lowest δ; the higher ω is contributed to their fast computation time. The MSER and SURF detectors with SIFT descriptors had some of the lowest ω even though they had some of the highest δ; these detectors also had a ω of zero for WFOV. The low ω was a result of the long computation times; the t_s was so long that the target could travel twice the range of the calculated δ before one and a half scans could be completed by these detectors. The reason for the STAR detector ω being zero was similar; the δ was zero for the smaller ship images, making the area under the curve zero for the longer ranges. The STAR method had the lowest ω for all FOVs as a result of having a zero δ for small ships.

Similarly, ORB also could not compute features on the smallest images, giving it a δ of zero at long ranges. However, having a low t_F allowed ORB to still have a large ω when calculated with weighted P_D.

Table 5. The 12 best BOW detectors ω calculated from δ for a 10-meter MHH contact with a relative velocity of 20 knots.

Keypoint	Descriptor	Matcher	Words	Sweep Width ω (km)			
				WFOV	MFOV	NFOV	UNFOV
SURF	SURF	L2	25	0.22	1.94	5.67	8.84
SURF	SURF	FLANN	50	0.22	1.95	5.68	8.82
SURF	SIFT	FLANN	25	0.00	1.55	4.58	5.13
SIFT	SURF	L1	25	0.19	1.82	5.34	8.42
SIFT	SURF	L2	25	0.19	1.82	5.33	8.38
FAST	SIFT	L2	50	0.21	1.82	5.30	7.79
STAR	SURF	FLANN	25	0.00	0.60	1.92	3.91
MSER	SIFT	L1	50	0.00	1.59	4.74	5.46
GFTT	SURF	L2	25	0.15	1.54	4.58	7.42
HARRIS	SURF	FLANN	100	0.20	1.75	5.14	8.07
ORB	SURF	FLANN	25	0.10	1.32	3.99	6.81
BRISK	SURF	L2	25	0.17	1.74	5.12	8.24

Methods that had fast computation times also benefited from the increase in detection probability from multiple glimpses. Analysis of the possible increase was considered by using weighted P_D as the upper bound ω and is displayed in Table 6. The weighted P_D increased the ω of many of the detectors over 40 percent for WFOV, 30 percent for MFOV and NFOV and 20 percent for UNFOV. The SURF key point SURF descriptor detectors are now surpassed by the BRISK and HARRIS detectors for NFOV and UNFOV; HARRIS also had the top ω for MFOV. The SURF feature SURF descriptor detectors ω increased for WFOV and maintained the highest ω for this FOV. Consequently, it was found that implementing pre-scaling increased δ by sacrificing computation time. The impacts on both the δ and t_F are considered through the evaluation method.

Table 6. The 12 best BOW detectors ω calculated from P_D for a 10-meter MHH contact with a relative velocity of 20 knots.

Keypoint	Descriptor	Matcher	Words	Sweep Width (km)			
				WFOV	MFOV	NFOV	UNFOV
SURF	SURF	L2	25	0.28	2.25	5.67	8.84
SURF	SURF	FLANN	50	0.28	2.26	5.68	8.82
SURF	SIFT	FLANN	25	0.00	1.55	4.58	5.13
SIFT	SURF	L1	25	0.25	1.82	5.34	8.42
SIFT	SURF	L2	25	0.25	1.82	5.33	8.38
FAST	SIFT	L2	50	0.21	1.82	5.30	7.79
STAR	SURF	FLANN	25	0.00	0.68	2.15	4.25
MSER	SIFT	L1	50	0.00	1.59	4.74	5.46
GFTT	SURF	L2	25	0.22	2.07	5.97	8.73
HARRIS	SURF	FLANN	100	0.27	2.30	6.60	9.40
ORB	SURF	FLANN	25	0.14	1.78	5.29	8.49
BRISK	SURF	L2	25	0.24	2.22	6.48	10.02

6. Improvements in Sweep Width by Pre-Scaling

Pre-scaling as a preprocessing stage increased the t_F and was only implemented on the BOW detectors that had a t_F of less than one second. The timing data was obtained using the previous method, with the exception of also pre-scaling by a factor of two the 1920 by 1080 images. The resulting t_F and t_s are displayed in Table 7. The t_F more than doubled for eight of the nine fastest detectors. ORB was the only detector where t_F increased less than the scaling factor, surpassing BRISK as the fastest detector. Shown in Table 8 is the number of glimpses resulting from the increased pre-scaling computation time. Previously, five methods had multiple glimpses for NFOV and UNFOV; now only the binary feature methods ORB and BRISK have multiple glimpses for these FOVs. Without pre-scaling all methods had multiple glimpses for WFOV; with pre-scaling only five methods have multiple glimpses for WFOV. Substantial gains in the δ for these detectors on small images overcame the increase in t_F and decreases in glimpses.

Table 7. The t_F and t_s of the nine fastest BOW detectors for two-times pre-scaling.

				t_F (seconds)		*Sweep time* t_s (seconds)			
Keypoint	Descriptor	Matcher	Words	Initial	2 x zoom	WFOV	MFOV	NFOV	UNFOV
SURF	SURF	L2	25	0.554	1.880	21	47	124	310
SURF	SURF	FLANN	50	0.615	2.017	22	50	133	333
SIFT	SURF	L1	25	0.959	3.626	40	90	239	598
SIFT	SURF	L2	25	0.959	3.628	40	90	239	599
STAR	SURF	FLANN	25	0.145	0.535	6	13	35	88
GFTT	SURF	L2	25	0.169	0.577	6	14	38	95
HARRIS	SURF	FLANN	100	0.167	0.571	6	14	38	94
ORB	SURF	FLANN	25	0.101	0.170	2	4	11	28
BRISK	SURF	L2	25	0.086	0.199	2	5	13	33

Table 8. Number of glimpses by FOV of the nine fastest BOW detectors for two-times pre-scaling.

				t_F (seconds)	*n* (number of glimpses)			
Keypoint	Descriptor	Matcher	Words		WFOV	MFOV	NFOV	UNFOV
SURF	SURF	L2	25	1.880	1	1	1	1
SURF	SURF	FLANN	50	2.017	1	1	1	1
SIFT	SURF	L1	25	3.626	1	1	1	1
SIFT	SURF	L2	25	3.628	1	1	1	1
STAR	SURF	FLANN	25	0.535	5	2	1	1
GFTT	SURF	L2	25	0.577	5	2	1	1
HARRIS	SURF	FLANN	100	0.571	5	2	1	1
ORB	SURF	FLANN	25	0.170	17	7	5	2
BRISK	SURF	L2	25	0.199	15	6	5	2

The δ results from pre-scaling are shown in Table 9 and were obtained from the ROC Curves in Appendix A, ROC Curves, Figure 33 through Figure 39. Both the STAR and ORB detectors no longer had a zero δ for any of the scaled images. Some of the δ also went down; most notably in the two SIFT keypoint SURF descriptor detectors for the largest three scales of ships. The two SIFT keypoint SURF descriptor detectors δ increased for the smaller scales. The ORB and BRISK detectors had the smallest increase in t_F and largest increase in δ for the smallest images. GFTT had an increase in δ for all image scales except full scale, which was unchanged. The increases in δ of GFTT propelled it to have some of the largest ω.

Table 9. The δ of the nine fastest BOW Detectors on evaluation image scales 100 percent to 10 percent for two-times pre-scaling.

Keypoint	Descriptor	Matcher	Words	100 (%)	75 (%)	50 (%)	25 (%)	20 (%)	15 (%)	10 (%)
SURF	SURF	L2	25	93	87	89	86	81	80	80
SURF	SURF	FLANN	50	93	92	92	86	80	79	79
SIFT	SURF	L1	25	82	82	80	86	89	84	84
SIFT	SURF	L2	25	88	82	85	85	91	84	84
STAR	SURF	FLANN	25	91	89	89	83	74	18	18
GFTT	SURF	L2	25	92	90	88	87	88	75	75
HARRIS	SURF	FLANN	100	90	80	78	78	78	71	71
ORB	SURF	FLANN	25	86	83	76	70	75	67	67
BRISK	SURF	L2	25	83	91	93	80	75	78	78

The ω calculated using δ for the nine fastest BOW detectors and two times pre-scaling are displayed in Table 10; the lateral range curves used to produce these ω are located in Appendix B, Lateral Range Curves, Figure 60 through Figure 63. Most of the detectors ω increased for the all FOVs. SIFT keypoint SURF descriptor L1 detector was the only method where ω decreased for UNFOV. Both SIFT keypoint SURF descriptor detectors had a decrease in ω for WFOV. Both SURF keypoint SURF descriptor detectors maintained the highest ω for WFOV. For MFOV, GFTT and the two SURF SURF detectors had the largest ω. GFTT surpassed all the detectors for the largest ω for NFOV and UNFOV. The BRISK detector ω also surpassed the SURF keypoint SURF descriptor detectors for UNFOV. The BRISK detector surpassed all other BOW detectors for all FOVs with ω calculated using the upper bound of P_D. Table 11 contains the ω calculated using P_D. The upper bound of ω for pre-scaling compared to no pre-scaling increased and decreased for many of the detectors. The ω for HARRIS decreased for all FOV, except WFOV. The ω increased in all FOVs for STAR, GFTT, ORB and

BRISK. The ω calculated thus far have been for a specific contact down to the height and relative speed; speed and MHH are considered the variables in the next section.

Table 10. Sweep width calculated with δ of the nine fastest BOW detectors utilizing two-times pre-scaling on a 10-meter MHH contact and relative velocity of 20 knots.

				Sweep Width (km)			
Keypoint	Descriptor	Matcher	Words	WFOV	MFOV	NFOV	UNFOV
SURF	SURF	L2	25	0.24	2.05	5.93	8.95
SURF	SURF	FLANN	50	0.24	2.05	5.94	8.98
SIFT	SURF	L1	25	0.18	2.02	5.85	8.35
SIFT	SURF	L2	25	0.18	2.04	5.92	8.45
STAR	SURF	FLANN	25	0.08	1.38	4.17	7.31
GFTT	SURF	L2	25	0.23	2.05	5.97	9.27
HARRIS	SURF	FLANN	100	0.22	1.88	5.50	8.53
ORB	SURF	FLANN	25	0.20	1.79	5.24	8.17
BRISK	SURF	L2	25	0.23	2.02	5.86	9.07

Table 11. Sweep width calculated with P_D of the nine fastest BOW detectors utilizing two-times pre-scaling for a 10-meter MHH contact and relative velocity of 20 knots.

				Sweep Width (km)			
Keypoint	Descriptor	Matcher	Words	WFOV	MFOV	NFOV	UNFOV
SURF	SURF	L2	25	0.24	2.05	5.93	8.95
SURF	SURF	FLANN	50	0.24	2.05	5.94	8.98
SIFT	SURF	L1	25	0.18	2.02	5.85	8.35
SIFT	SURF	L2	25	0.18	2.04	5.92	8.45
STAR	SURF	FLANN	25	0.10	1.55	4.17	7.31
GFTT	SURF	L2	25	0.29	2.32	5.97	9.27
HARRIS	SURF	FLANN	100	0.28	2.21	5.50	8.53
ORB	SURF	FLANN	25	0.28	2.36	6.76	9.58
BRISK	SURF	L2	25	0.30	2.44	7.03	10.22

7. The Impact of Ship Size and Speed on Sweep Width

Ship size in pixels was set as the dependent variable in the evaluation model allowing for consideration of different physical MHH and contact speeds as an independent variable. The detection range was dependent on the size of the ship under consideration. The impact of MHH on ω is shown in Figure 9 for the BRISK model detector with two times zoom implemented. The solid lines are ω calculated using the weighted P_D providing the upper bound, and the dashed lines are the ω calculated using a single δ per sweep. In Figure 9, the concavity of ω increases for MFOV, NFOV and UNFOV beginning at five kilometers influenced by the horizon. It is also observed from Figure 9 that a contact with a MHH of less than six meters cannot be detected when using the WFOV lens.

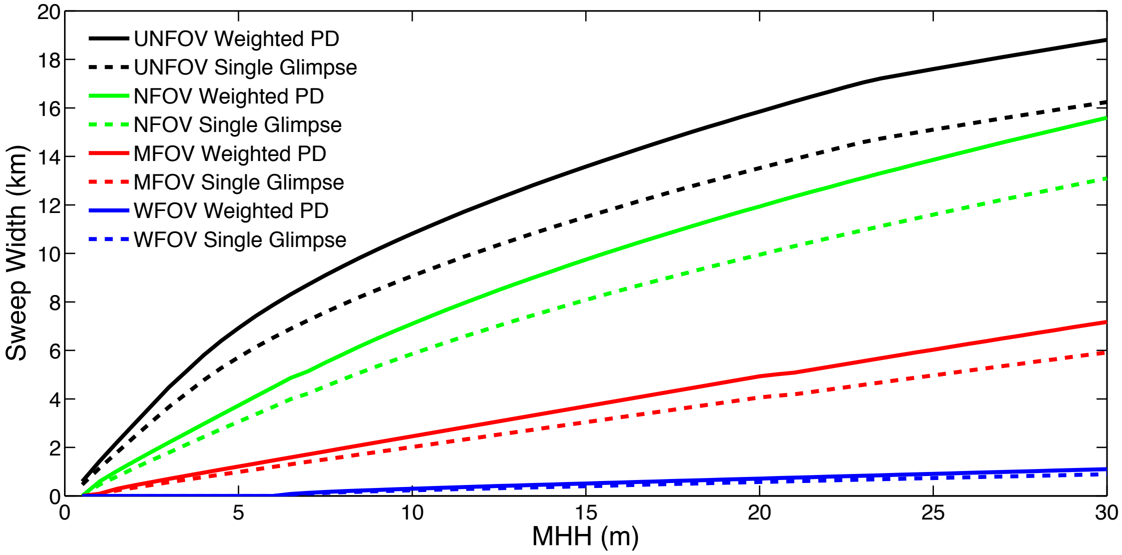

Figure 9. Graph of ω by MHH for the BRISK BOW detector with two-times pre-scaling.

The impact of a contact's velocity on the ω for a 10-meter MHH contact is displayed in Figure 10. The ω goes down as the relative speed between the sensor and the contact goes up. The solid line again represents the upper bound of ω calculated by P_D, and the dashed line is ω calculated with a single δ per sweep. The speed of the contact had less of an impact on detection range for this BRISK BOW two times zoom detector over slower computational detectors. The BRISK BOW detector swept at the imposed hardware limited t_s, allowing the contact of interest to only travel a short distance between sweeps. This short t_s did not prevent the 10-meter MHH contact from being undetected by a WFOV when traveling faster than 34 knots.

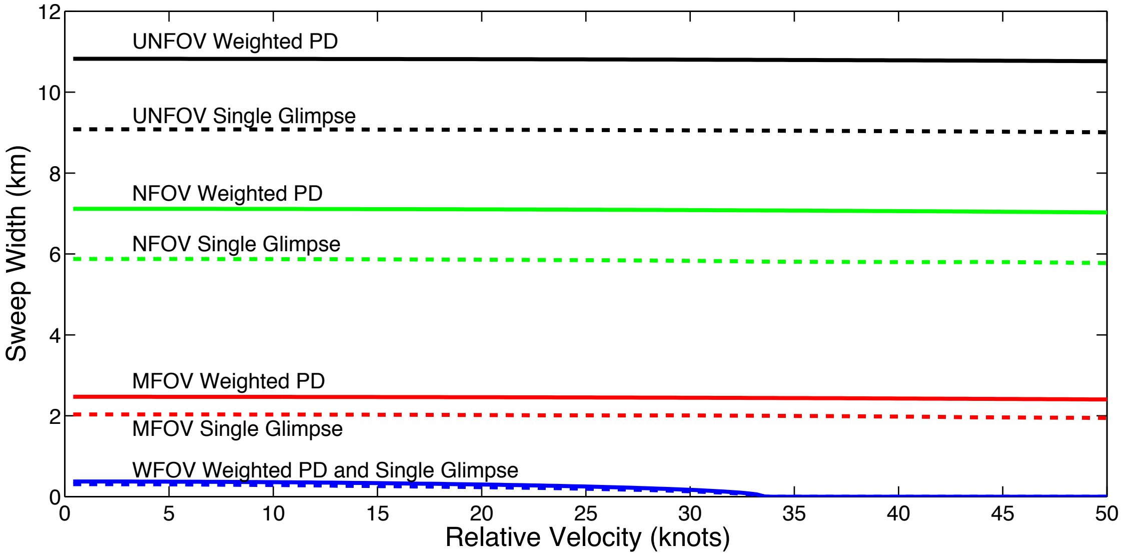

Figure 10. Graph of ω by relative velocity for the BRISK BOW detector with two-times pre-scaling.

D. HISTOGRAMS OF ORIENTED GRADIENTS

The HOG detector was built similar to the original work "Histograms of Oriented Gradients for Human Detection" [4]. The design of experiments approach was used to create over 3,000 ship detectors out of HOG descriptors. The size of the descriptors and manipulation of the training images were used in the design of experiments. A systematic selection approach was used to reduce the large number of detectors to a few for the evaluation method. The OpenCV libraries were utilized to implement the same detection of ships with the HOG descriptors on a GPU reducing the computation time. Mixed results were obtained in the comparison of the δ between the CPU and GPU implementation. Pre-scaling was attempted to increase the δ. The final internal HOG comparison was done on seven descriptors implemented on a GPU.

1. Establishing Training and Design of Experiments for HOG

Prior to the design of experiments there were many trial-and-error implementations of training HOG descriptors. As described above in Chapter II.E, the HOG detector used a sliding window approach; the window that is scanned is a HOG descriptor. Displayed in Figure 11 is a visualization of a HOG descriptor. In Figure 11 the length of the line displays each bin magnitude, and the direction of the line corresponds the gradient angle forming the bin. The initial HOG implementation took the entire set of positive and negative training images and transformed them into HOG descriptors. Then, all of these descriptors were applied to a linear SVM to compute the representation of the positive HOG descriptor for matching. The first improvement found was in creating two descriptors from every training image, one from the original image and one from the image flipped over the *y*-axis. The flipping created a more symmetric HOG descriptor and was not biased when the training set had more port aspects of ships than starboard aspects of ships and vice versa. The descriptor in Figure 11 was trained using flipped descriptors and is not entirely symmetrical, though some symmetry is observed.

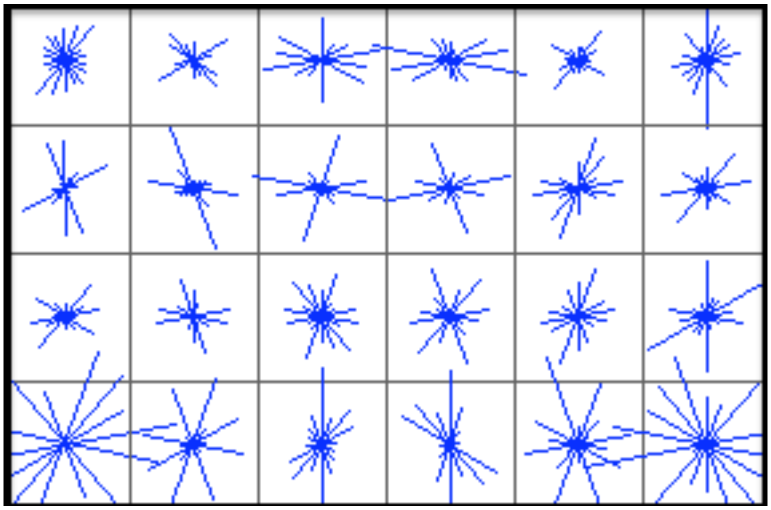

Figure 11. Visualization of a 32 pixels tall by 48 pixels wide HOG descriptor.

The next improvement came from the implementation of a bootstrapping approach for the negative training images. P. M. Roth et al. provide a basis for improving training with bootstrap learning [46]. The bootstrapping approach implemented selected a percentage of the negative images to not initially train on and then created a detector from the descriptors that had been used for training to that point. Once a detector was trained, the initial trained detector was evaluated on the percentage of negative images left out for bootstrapping. The positive detections during this process were then added to the negative training descriptors, both the original and the flipped descriptors. Initially, the set of trained images were all scaled down to fit in the descriptor, whereas the bootstrapped negatives used a sliding window that could return many false positives per bootstrap image. Even though there were only 105 negative images for training, the sliding window could find thousands of false positives while bootstrapping. When the negative descriptors greatly outnumbered the positive descriptors, the detector returned by the SVM also favored the negative descriptors by returning more false positives. Limiting the number of negative training descriptors to three times the positive training descriptors prevented some of the biasing. The best detectors found a very small number of false positives to bootstrap because they already had developed a good descriptor prior to bootstrapping.

The HOG detection descriptors produced in training had a fixed height and width. The height and width were constrained to be multiples of eight and each eight by eight pixels square created one histogram. All descriptors used a constant nine-bin histogram. The bin size and number of bins were constrained to the allowed values of the GPU implementation. The height and width of the descriptors were varied in the design of experiments, always as a multiple of eight. The training images were resized to the size of the descriptors, not preserving image ratio. The mean ship height to width ratio of the cropped training images was 1 to 2.16. The descriptor heights were varied from 32 to 188 pixels and widths from 40 to 288 pixels. The bootstrapping percentage was varied from zero percent to 40 percent while limiting the number of negative descriptors to three times the number of positive descriptors. Many of the HOG detectors produced a δ of greater than 90 percent on the full-scale images.

2. Selecting the Best HOG Detectors

The small images were very challenging for larger descriptors. Many of the large descriptors had the highest δ for full scale images but required upwards of seven times pre-scaling to detect any ship in the smallest scale images. The small detectors could detect ships in the small images without pre-scaling. Not requiring pre-scaling was found to be important in the evaluation model. The small descriptors could still easily detect larger ships by the scale down and rescan method of multiscale detection. The smaller descriptors were also faster computationally, requiring fewer comparisons between the descriptor bins and the image bins. The findings showed that the height and widths did not need to be near the average ratio of the training images or evaluation images in order to create a descriptor with a high δ. The best HOG descriptors were selected following an approach similar to selection of the BOW detectors. All HOG descriptors were evaluated on all scales of images with multiple threshold values, and 10 detectors from each scale were selected based on δ. Seven detectors were selected to evaluate and compare to the other detection methods; detectors were in the top 10 δ for multiple scales.

3. Optimization Through Pre-Scaling and Graphical Processors Computation

Improvements for the selected descriptors, δ and t_F, were attempted through pre-scaling and implementation on a 192 CUDA core GPU. The δ were successfully improved through pre-scaling. The byproduct of pre-scaling was that t_F was increased by more than a factor of four when pre-scaled by a factor of two. The t_F were reduced by implementation of the HOG detector on the GPU but not enough to overcome the factor of four increase from pre-scaling. Nearly all computation was now done on the GPU, and only the resulting locations of positive detections were returned. The image memory transfer time was included in the calculations. Initial timing test for a single image showed the time to transfer the image was over two seconds, whereas the computation time was a fraction of a second. The two seconds was found to be the initial transfer time for the first image and initial setup of the GPU for computation. All transfer times after the first image were fractions of a second. The initial setup time was removed by pre-loading an additional image onto the GPU before starting the timing test on the batch of 200 images.

The OpenCV library GPU methods were limited to performing detections on grayscale images vice the RGB images that were initially used on the CPU version of the HOG detector. All testing was then conducted on grayscale images for HOG after discovering that loading the images as grayscale had little impact on the δ and reduced the t_F. Overall, the best ω was from the GPU implementation of the detector with no pre-scaling. The next best implementation was the CPU detection with no pre-scaling, followed by the GPU detector with two times pre-scaling, and last the CPU detection with two times pre-scaling. Next, the CPU and GPU versions of the seven best HOG detectors are compared with no pre-scaling through the evaluation method.

4. Evaluation of Glimpse Probability for the Top HOG Detectors

The δ for the CPU and GPU implementation are similar but not identical. In some cases the CPU outperformed the GPU, and in some cases the GPU outperformed the CPU. The δ for the top seven HOG detectors evaluated using the CPU are given in Table 12 and evaluated using the GPU are in Table 13. The ROC curves used to obtain these δ are provided in Appendix A, ROC Curves, Figure 40 through Figure 53. The slight differences between the computation done with the CPU and the GPU are observed when comparing Table 12 and Table 13. Neither the GPU nor CPU showed a single detector that was superior to the rest using δ. These detectors also had very fast, yet similar t_F and number of glimpses.

Table 12. The δ of the top seven BOW detectors for evaluation image scales 100 percent to 10 percent performed on CPU.

Height	Width	Bootstrap (%)	100 (%)	75 (%)	50 (%)	25 (%)	20 (%)	15 (%)	10 (%)
32	48	5	94	93	88	81	81	74	57
32	48	15	91	94	86	81	81	79	60
32	48	25	95	92	94	86	86	76	60
32	48	30	95	90	92	84	84	78	68
32	48	40	94	94	88	84	84	73	66
32	56	30	96	93	93	87	87	77	42
32	56	40	98	91	89	84	84	77	44

Table 13. The δ of the top seven BOW detectors for evaluation image scales 100 percent to 10 percent performed on GPU.

Height	Width	Bootstrap (%)	100 (%)	75 (%)	50 (%)	25 (%)	20 (%)	15 (%)	10 (%)
32	48	5	95	94	88	86	86	73	61
32	48	15	88	93	90	85	82	79	59
32	48	25	95	96	91	89	83	80	59
32	48	30	96	93	93	86	82	78	67
32	48	40	97	93	87	88	80	73	63
32	56	30	93	86	94	86	82	78	42
32	56	40	95	95	93	88	84	77	52

5. Frame Time, Sweep Time and Number of Glimpses for CPU and GPU HOG Detectors

The top seven HOG descriptors all had a t_F of under 0.2 seconds, enabling them all to have multiple glimpses for all FOVs. The GPU was only slightly faster than the CPU, computing at 66 percent to 72 percent of the time per frame. The CPU and GPU t_F and the number of glimpses for the GPU are displayed in Table 14. The calculated t_F was obtained using the same method as before. The HOG detector provides more information than the BOW detector does on a positive detection. The HOG detector provides the location in the image of the positive detection or localizes the detection in the image. The HOG descriptor can also have multiple positive detections in the positive and negative images. To compare the BOW, HOG and DPM, only a single positive detection was considered in the positive and negative images. The HOG detector also performs multi-scale detection bounding small ships as well as large ships. The amount of scaling and number of levels to scale were implemented as runtime variable. All of the evaluations for determining δ and t_F were obtained with a scale factor of 1.25 and 12 levels of scaling. Optimization was not conducted for the scale factor or levels of scales and could be done to improve the detectors even more. The selected scale factor and levels of scales covered all the evaluation image sizes and was selected through a limited trial-and-error approach.

Table 14. The t_F of the top seven HOG detectors performed on both CPU and GPU and the number of glimpses from the GPU by FOV.

Height	Width	Bootstrap (%)	CPU t_F (seconds)	GPU t_F (seconds)	GPU Glimpses			
					WFOV	MFOV	NFOV	UNFOV
32	48	5	0.173	0.116	25	11	8	3
32	48	15	0.189	0.135	22	9	7	2
32	48	25	0.197	0.142	21	9	7	2
32	48	30	0.176	0.120	25	11	8	3
32	48	40	0.175	0.122	24	10	8	3
32	56	30	0.184	0.125	24	10	8	3
32	56	40	0.176	0.117	25	11	8	3

An example of the localization of multiple ship detections is shown in Figure 12. The image in Figure 12 was not part of the evaluation set but contains multiple sailboats, and the HOG ship detector was able to detect them well. Notably, the training set contained no sailboats. All positive detections are boxed in green in the image. There are two ships that have multiple positive detections and were not suppressed from grouping by the multiscale detection. An additional run time variable was the number of scales to group if there are multiple detections in the same region of the image but at different scales. The two boxes that overlap near the right side of the image are not counted as two detections on the same ship since there are two sailboats there. The 32 pixel tall by 48 pixel wide HOG descriptor that was created using 30 percent of the negative images to bootstrap produced the detections in Figure 12. This HOG descriptor is displayed in Figure 11 and had the highest ω when calculated using both weighted P_D and δ.

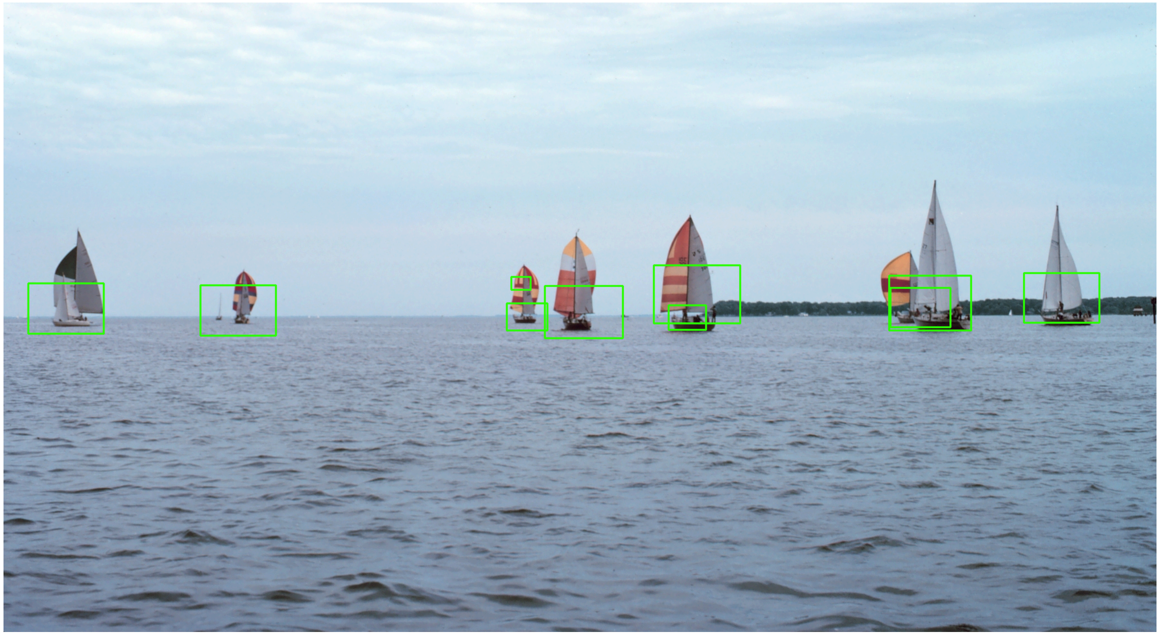

Figure 12. Example of multiple ship detection with localization using HOG detector (after [39]).

6. Lateral Range Curves and Sweep Width of HOG Detectors

The lateral range curves are provided for the top seven HOG descriptors evaluated using the GPU and calculated with δ in Appendix B, Lateral Range Curves, Figure 64 through Figure 67. From the trapezoidal approximation of the area under these curves, the values of ω were obtained. The ω calculated using the δ alone and P_D are shown in Table 15 and Table 16. In the final stage of the evaluation, there was one HOG descriptor that stood out having the highest ω. Also provided is the ω by MHH and ω by relative velocity in Figure 13 and Figure 14. Observed from these figures are similar effects as discussed for the BOW detector. The ω becomes concave down for contacts over the horizon in Figure 13. In Figure 14, the ω for HOG goes down slower as relative velocity increases compared to BOW. The slower decrease is contributed to the faster computation time of the HOG method on a GPU. Both graphs again show a zero detection probability when using WFOV. The zero detection probability is for very small contacts that are close to the sensor in Figure 13 and very fast contacts in Figure 14.

Table 15. The ω calculated using δ from the top seven HOG detectors performed on a GPU for a 10-meter MHH contact with relative velocity of 20 knots.

Height	Width	Bootstrap (%)	Sweep Width ω (km)			
			WFOV	MFOV	NFOV	UNFOV
32	48	5	0.21	1.97	5.78	9.12
32	48	15	0.22	1.98	5.79	9.10
32	48	25	0.22	2.03	5.92	9.32
32	48	30	0.22	2.03	5.94	9.32
32	48	40	0.21	1.97	5.76	9.09
32	56	30	0.20	1.92	5.63	8.99
32	56	40	0.21	1.98	5.81	9.22

Table 16. The ω calculated using P_D from the top seven HOG detectors performed on a GPU for a 10-meter MHH contact with relative velocity of 20 knots.

Height	Width	Bootstrap (%)	Sweep Width ω (km)			
			WFOV	MFOV	NFOV	UNFOV
32	48	5	0.28	2.40	6.94	10.46
32	48	15	0.28	2.40	6.94	10.19
32	48	25	0.28	2.40	6.95	10.29
32	48	30	0.29	2.44	7.04	10.58
32	48	40	0.28	2.41	6.96	10.45
32	56	30	0.25	2.29	6.66	10.23
32	56	40	0.27	2.36	6.83	10.40

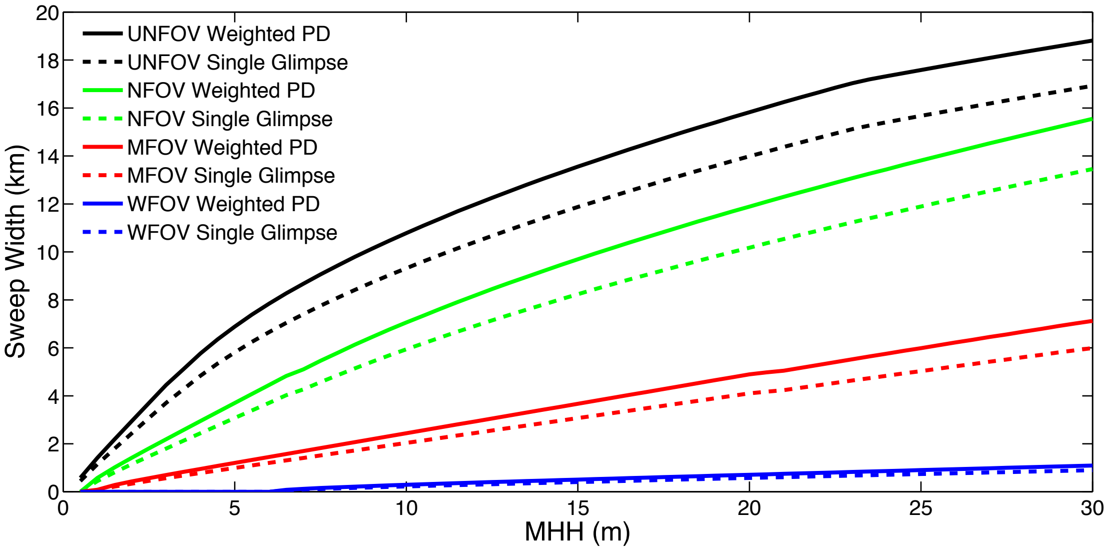

Figure 13. Graph of ω by MHH of contact from the top HOG ship detector.

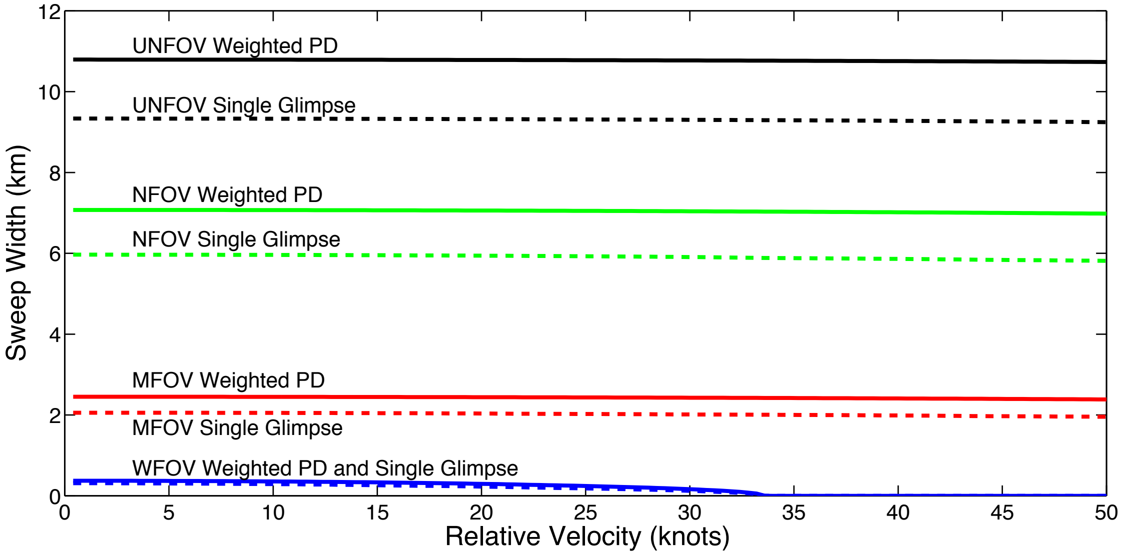

Figure 14. Graph of ω by relative speed from the top HOG ship detector for a 10-meter MHH contact.

E. DEFORMABLE PARTS MODEL

The DPM detector was trained using a modified version of the D. Ramanan and Y. Yang flexible mixture of parts original training algorithm [27]. The detection program for evaluation was developed with a DPM OpenCV library created by H. Bristow [26]. The DPM training was semi-supervised, requiring labels for the center of each part and the tree structure connecting the parts. The DPM ship detector design of experiments used four variables: the number of parts, connections of parts, allowed overlap of parts and HOG descriptor parameters.

1. Design of Experiments for DPM

Over 300 ship detectors were successfully trained in the DPM design of experiments. Thousands of combinations were attempted in the design of experiments, but the majority of attempts were unsuccessful in creating a detector from the training set. The number of parts was varied from three to 20. The parts tree structure was initially varied and subsequently found to produce the best results with the tree starting on a mast or on the superstructure. The number overlapping parts was varied from no overlapping parts to all parts allowed to overlap. The DPM HOG descriptor bin size and number of bins was also varied from four to 15. The training method used 20 percent of the images for bootstrapping and was not changed. The HOG descriptors used for DPM dynamically changed during training vice being set as the descriptors from the HOG detector.

The DPM HOG features were limited to square window, and the window sizes were increased during training until all parts for all images met a desired descriptor threshold. There were also four poses created for each DPM detector. The DPM training failed if it required the creation of more than four poses to describe the training set. In Figure 15 a seven-part DPM model is shown and is one of the four poses trained for this model. All four descriptors for this seven-part DPM are shown in Figure 16. This descriptor was trained on full size training images and is why each descriptor is so large. The magnitude of the gradient in these descriptors is displayed by the intensity of the white instead of the length of the vector as before with the HOG descriptor. The bow, stern, superstructure and mast can be seen in the descriptor shown in Figure 15.

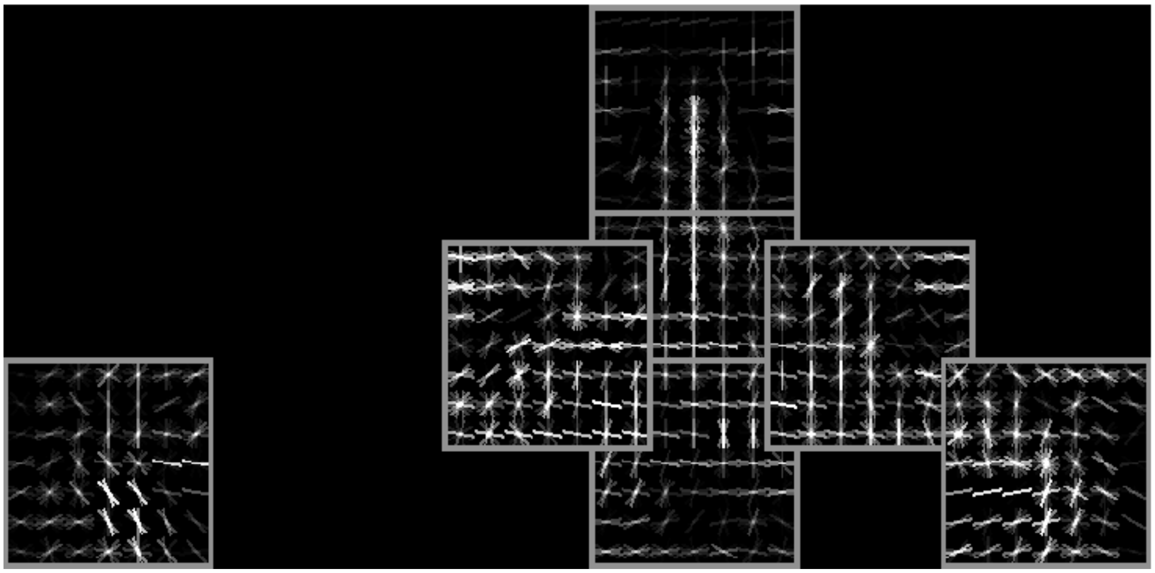

Figure 15. Visualization of a single pose DPM descriptor from a seven-part model nine bin HOG.

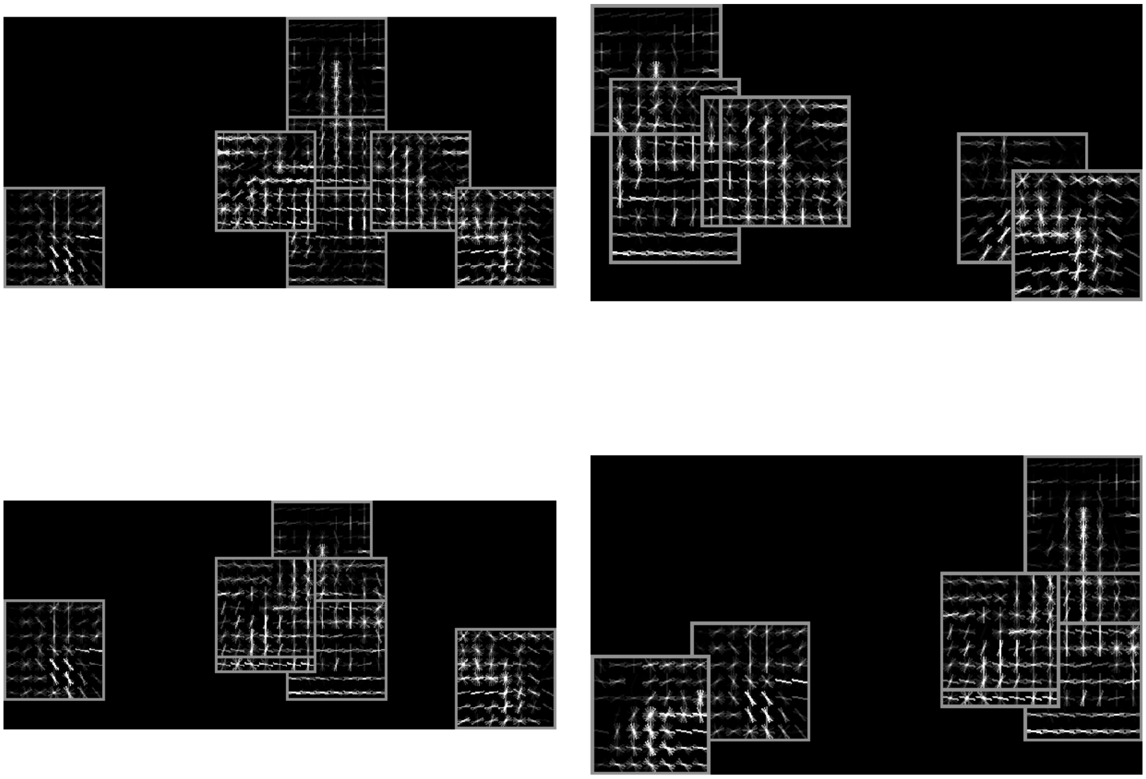

Figure 16. Visualization of the four poses of a seven-part, nine bin HOG DPM descriptor.

This DPM was originally implemented as a pose detector, and each pose is shaped like a different class or aspect of ships as seen in Figure 16. The two models on the right more closely resemble a cargo or tanker ship, where the superstructure is located far away from the bow. The two descriptors on the left resemble vessels where the superstructure is located closer to amidships. Even though none of these models appear to cover ships where the bow or stern is facing the sensor, they can still detect ships with a narrow aspect as seen in Figure 17. Displayed in Figure 17 is a four-part model and seven-part model detection of the same vessel. The seven-part model does not need all parts on the ship to find a positive match. It appears that the descriptor positions in the upper left of Figure 16 was the pose used to detect the ship in the seven-part model of Figure 17. The visualization of the descriptor and the detection do not overlap exactly, demonstrating the uncertainty in position of the descriptors that the DPM allows. Only one of the four poses of the descriptor was needed to find a positive match as shown in Figure 17. Having these four different poses benefits the DPM detector and allowed it to have the highest δ thus far on the full size images. As with the HOG detector, DPM provided the location of detection in the images. For comparison, only a single positive detection was considered for each image.

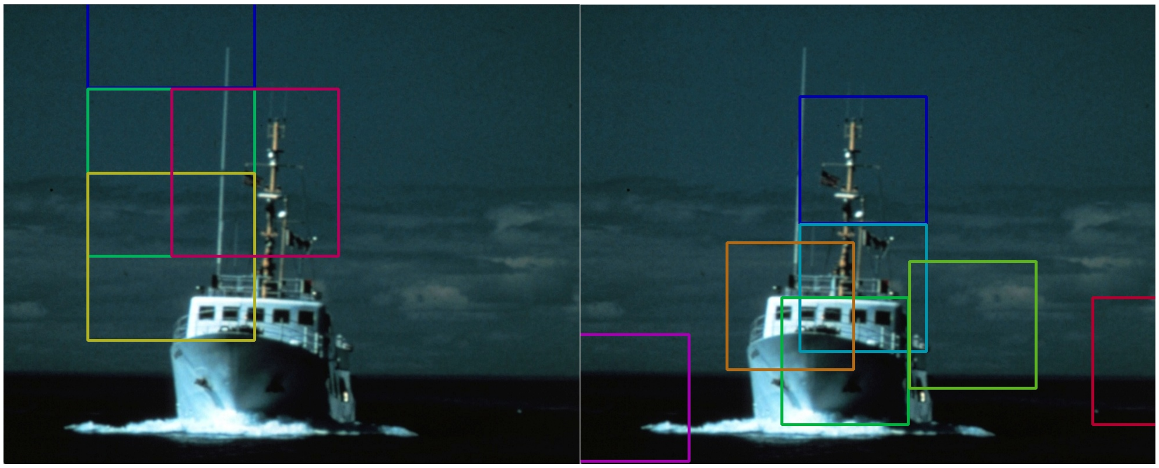

Figure 17. Visualized detection of a ship from a four-part model, left, and seven-part model, right (after [39]).

2. Selecting the Best DPM Detectors

Many of the DPM detectors performed extremely well on the full size images, with recalls above 98 percent while maintain a false positive percentage below five percent. The challenge that the DPM had was in detecting small vessels. Many of the DPM detectors could maintain very high recall and precision down to the 50 percent scaled images, but below the 50 percent scale recall quickly dropped to zero. The recall dropped to zero when the models were larger than the images. Shown in Figure 18 is the ROC curve for the seven-part DPM visualized in Figure 16. This seven-part model had highest δ out of all DPM detectors for full size images down to 50 percent scale images but then dropped to zero on smaller images. In Figure 18, when the 25 percent scale curve goes horizontal, the positive detections reached 100 percent for the evaluation images that were large enough for the detector to detect a ship. Pre-scaling was not found as successful as with other methods.

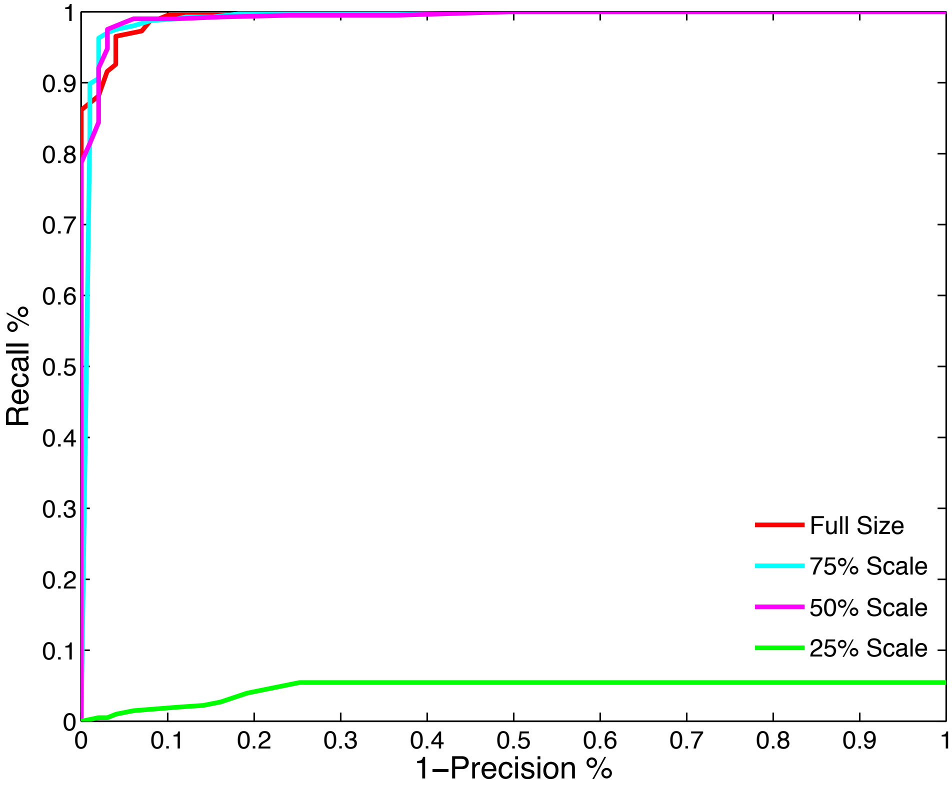

Figure 18. ROC curve for seven-part DPM detector on full size to 25 percent scale evaluation images.

After applying two-times pre-scaling to the images, this seven-part model could detect ships in the 25 percent and 20 percent scales better than any other DPM detector. The issue again was that this DPM detector could not detect ships in the smaller images. The ROC curve for this seven-part detector with two-times pre-scaling for full scale to 10 percent scaled images is displayed in Figure 19. Using the fact that the DPM detector could successfully detect ships that were one-half the size of the original training images, training was attempted using 20 percent scaled training images instead of trying to pre-scale by a factor of four.

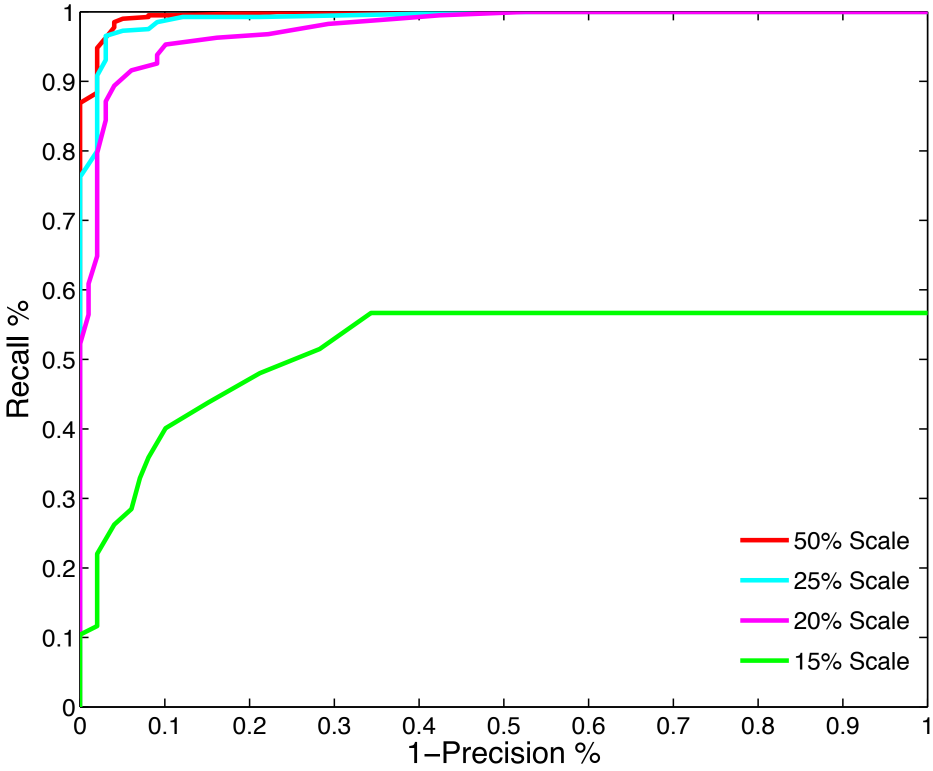

Figure 19. ROC curve for seven-part DPM detector on full scale to 10 percent scaled evaluation images when using two-times pre-scaling.

Training with 20 percent scaled images was not successful; no detector model had a δ of greater than 90 percent for any scale. Training on 25 percent scaled positive images had successful results. DPM detectors that could detect ships in the 15 percent scale images without the need for pre-scaling were created; these detectors still failed to detect ships in over half of the 10 percent scaled images. The DPM model with the best resulting δ was a four-part model, and a visualization of this model is shown in Figure 20. The model creates four poses of the parts, though all four poses were identical as shown from the visualization. The four parts of this four-part model also all overlap, making the DPM descriptor in essence a single HOG descriptor. Only three parts are shown for the four-part model since one of the parts was entirely covered by the other parts. Computation time was another challenge for the DPM detector.

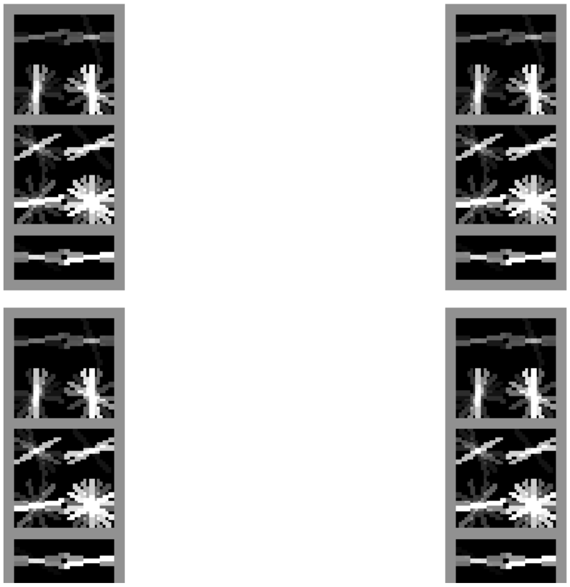

Figure 20. Visualization of the four poses of a four-part, nine-bin HOG DPM descriptor.

3. Frame Time and Sweep Time for DPM

Though DPM has some of the highest δ for large ships, it comes at the cost of long computational time. The t_F for both the seven-part model and the four-part model was two seconds, which was faster than any of the BOW models that used SIFT descriptors vice SURF descriptors. A t_F of two seconds was not fast enough to have multiple glimpses, even with the WFOV lens, but two seconds was fast enough to meet the maximum rotation time of 30 seconds with the WFOV lens. When pre-scaling was used, t_F did not increase linearly with the pre-scale factor, and a quadratic increase was a closer approximation. For a pre-scale factor of two, the t_F increased to 7.6 seconds and increased to 34 seconds for a pre-scale factor of four. Pre-scaling by a factor of four would be required for the seven-part model to detect ships of sizes represented by the 10 percent scaled images. Shown in Table 17 are the t_F and the associated t_s for the DPM detectors.

Table 17. The t_F and t_s for the DPM detectors by FOV.

	t_F (seconds)	*Sweep time* t_s (seconds)			
		WFOV	MFOV	NFOV	UNFOV
No pre-scaling	2.0	30	49.5	132	330
Two times pre-scaling	7.3	80	180	482	1,204
Four Times pre-scaling	34.0	374	842	2,244	5,610

NFOV and UNFOV had very long t_s. Without pre-scaling in UNFOV, the DPM detector took five and a half minutes to complete an entire sweep. With two times pre-scaling, the t_s increased to more than 20 minutes. Implementation of four times pre-scaling would increase a single sweep to more than an hour and a half. There has been some significant improvements to DPM detectors recently with the iRobot at Nvidia GPU Technology Conference 2013 showing a DPM that was five times faster when performed on a GPU and CPU vice a CPU alone [47]. The results of this DPM model gave a t_F of less than 200 milliseconds when performing on VGA images, which are 640 by 480 pixels. This is less than one sixth the pixel area that was used to calculate t_F in this evaluation.

Table 18. The δ of the DPM detectors for evaluation image scales 100 percent to 10 percent.

Detector Model	100 (%)	75 (%)	50 (%)	25 (%)	20 (%)	15 (%)	10 (%)
Four-part Model no pre-scaling	96	91	93	94	92	93	40
Seven-part Model no pre-scaling	97	96	99	50	0	0	0
Seven-part Model two time pre-scaling	97	98	99	97	92	57	0

4. Lateral Range Curves and Sweep Width of DPM Detectors

The four-part model with no pre-scaling and the seven-part model with and without two-times pre-scaling were selected to calculated the ω. These were the two best

DPM detectors; one was produced from small training images and one from large training images. The δ for these models are shown in Table 18. The δ were calculated using the ROC curves shown in Figure 18 and Figure 19 for the seven-part model and from Figure 54 in Appendix A, ROC Curves, for the four-part model. None of these detectors had a t_F low enough to have multiple glimpses at any FOV resulting in $P_D = \delta$ in Table 18. The lateral range curves for these three detection methods are provided in Appendix B, Lateral Range Curves, Figure 68 through Figure 71. The ω for these detectors are shown in Table 19. The four-part model had the largest ω for all FOVs. The seven-part model with two times pre-scaling had a lower ω with the UNFOV than with the NFOV and is the only detector that showed this phenomenon. The drop in ω was caused by the large increase in t_F and subsequently t_S. The seven-part model with and without pre-scaling has a ω of zero for WFOV resulting from the distance the 20-knot contact could travel was greater than twice the detectable range for these detection methods. Given that there are faster DPM implementations, ω was calculated for a five times faster t_F and the resulting ω values are shown in Table 20. For the case of a five times faster $t_{F,}$ multiple glimpses are possible and were considered for calculating the corresponding ω with P_D. The DPM methods did not break 10 km for the ω even with a five times faster detection rate, as the BOW and HOG detectors had when using P_D to calculate ω.

Table 19. The ω of the DPM detectors calculated using $P_D = \delta_,$ evaluated for a 10-meter MHH contact with a relative velocity of 20 knots.

Detector Model	Sweep Width (km)			
	WFOV	MFOV	NFOV	UNFOV
Four-part Model no pre-scaling	0.22	2.05	6.00	9.26
Seven-part Model no pre-scaling	0.00	0.60	1.93	3.42
Seven-part Model two time pre-scaling	0.00	0.97	3.24	1.60

Table 20. The ω of the DPM detectors calculated using P_D assuming a five times faster $t_{f,}$ evaluated for a 10-meter MHH contact with a relative velocity of 20 knots.

Detector Model	Sweep Width (km)			
	WFOV	MFOV	NFOV	UNFOV
Four-part Model no pre-scaling	0.25	2.27	6.54	9.62
Seven-part Model no pre-scaling	0.00	0.71	2.22	4.37
Seven-part Model two time pre-scaling	0.15	1.71	5.10	8.52

5. Sweep Width by Mast Head Height and Velocity

The ω by MHH and ω by relative velocity for the four-part DPM detector are provided in Figure 21 and Figure 22. The graph of ω by MHH was very similar to the graphs for the other model detectors; it differs in that there is only one line per FOV. For the DPM detector, the t_F was not fast enough to have multiple glimpses, leaving only a single curve per FOV. The longer t_F for the DPM also had a large and noticeable impact on the ω by relative velocity. The ω decreased much faster for the DPM detector method as relative velocity increased, especially for the UNFOV.

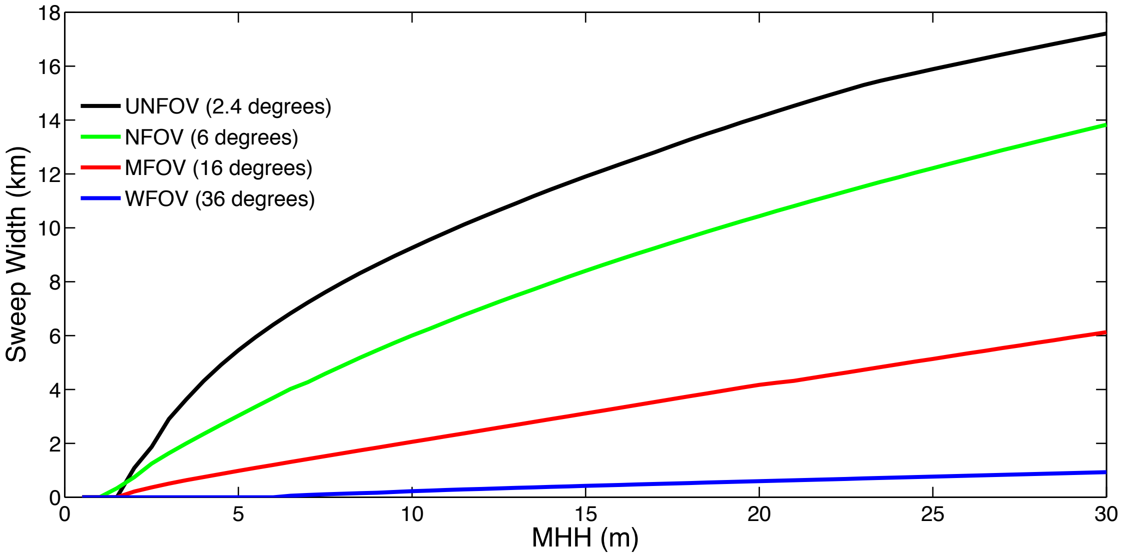

Figure 21. Graph of ω by MHH for the four-part DPM ship detector on a 20-knot relative velocity contact.

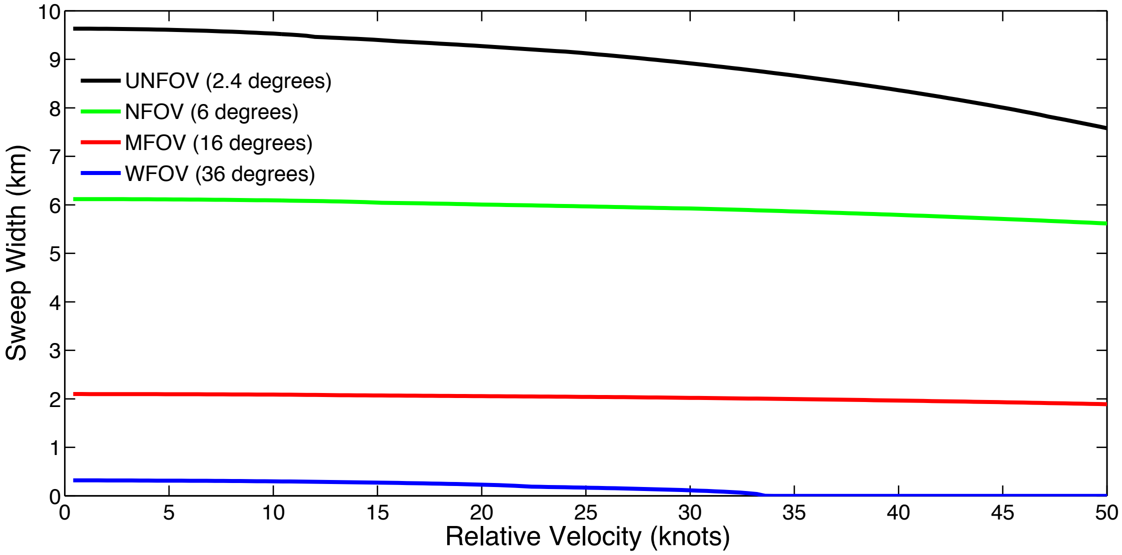

Figure 22. Graph of ω by relative velocity for the four-part DPM ship detector on a 10-meter MHH contact.

IV. RESULTS

In Chapter III, the evaluation method was created using search theory principles. The evaluation method models computer vision algorithms as a ship detector for operational naval platforms. Three object detection methods were then trained through experiments to optimize the algorithms for ship detection. Thousands of detectors were trained and evaluated. The evaluation of the best ship detector for each method is discussed in this section through the observations of these evaluations. A HYBRID detector was constructed using the HOG to find ROIs and the BOW method to determine if the ROIs contain a ship. The results of the three ship detection methods discussed in Chapter III as well as the HYBRID detector are compared in this chapter.

A. HYBRID HOG AND BOW

The HYBRID combines some benefits of the HOG and BOW methods. The BOW method has the highest glimpse probability δ for the smallest scale images. The HOG was the fastest detection method that could localize the detection in the images. The majority of the computation for HOG was processed in parallel on a GPU. The approach assumed that by turning the threshold down on the HOG detector, it would pass more ROI to be evaluated by the BOW method. The threshold settings for both the HOG and BOW methods were important. Low threshold values for HOG caused too many ROIs to be passed, causing the whole image to be evaluated by the BOW. If the HOG only passed true positive detections, then the detector would be no better than HOG alone.

The threshold values also had a large impact on the computational time. As the number of ROI that HOG detected increased, so did the computation done by the BOW model. The best results for the BOW occurred when two times pre-scaling was implemented; the HYBRID was also most successful when two times pre-scaling was implemented on the ROI. The highest sweep width ω for the HYBRID model resulted from using two-times pre-scaling with the HOG on the GPU. Many of the top BOW models were attempted with the HYBRID, and the greatest success was from SURF keypoints, SURF descriptor and L2 matcher method. The 32 by 48 HOG created using 30 percent bootstrapping found the ROI for the HYBRID. Reducing the threshold for the HOG detector made the grouping of rectangles done by HOG insufficient by itself, passing multiple ROI around the same ship. Implementing merging of these overlapping rectangles produced much better results and was added to the HYBRID detector.

B. COMPARITSON OF DETECTION METHODS

Comparisons were made between the top ship detectors from each method utilizing each stage of the evaluation model. In addition to the comparison of δ, the false positive rate was also compared. Challenges for calculating the computation time of the HYBRID method are discussed, along with computational time comparisons. The final stage of comparison was done using the ω for UNFOV with multiple MHH and contacts speeds.

1. Glimpse Probability and False Positive Rate

The best of the BOW, HOG and DPM based on the ω were selected to compare against the HYBRID implementation of HOG and BOW. The BOW with the largest ω was the BRISK SURF L2 25 word model using two-times pre-scaling. The HOG with the largest ω was the 32 by 48 model trained with 30 percent bootstrapping. The DPM with the largest ω was the four-part model trained on 25 percent scaled training images. The HYBRID method used the best HOG based on ω and used the SURF feature, SURF descriptor, L2 matcher and 25 word model that was in the best of the BOW models. Table 21 contains the δ for the top detectors from each method, and the ROC curves for the HYBRID detector are in Appendix A, ROC Curves, Figure 55. The δ of the HYBRID method did improve over the HOG method in the smaller scale images and in some cases surpassed the BOW and HOG methods.

Table 21. The δ of the top detectors from each method on evaluation image scales 100 percent to 10 percent.

Detector Method	100 (%)	75 (%)	50 (%)	25 (%)	20 (%)	15 (%)	10 (%)
BOW	83	91	93	80	75	78	78
HOG	96	93	93	86	82	78	67
DPM	96	91	93	94	92	93	40
HYBRID	94	96	93	92	87	86	76

An observation made through comparisons of the ROC curves was that even though obtained δ did not improve much in the HYBRID method, the false positive percentage was greatly reduced. The false positive probabilities that correspond to the calculated δ are displayed in Table 22. The BOW and HOG models had a false positive probability of greater than 30 percent for their calculated δ on the 10 percent scale images, where the HYBRID method had 15 percent false positive detections at the calculated δ for this scale. The DPM method had the lowest false positives and highest δ for 25 percent down to 15 percent scaled images but then had the lowest δ on the smallest scale ships with a higher false positive rate. The DPM and HYBRD methods did very well on the 50 percent to full-scale images, and both had less than a 10 percent false positive rate for these scales. The HYBRID method had the lowest average false positive rate and surpassed the DPM method in computational speed.

Table 22. False positive probabilities of the top detectors from each method on evaluation image scales 100 percent to 10 percent.

Detector Method	100 (%)	75 (%)	50 (%)	25 (%)	20 (%)	15 (%)	10 (%)
BOW	15	18	18	22	17	22	43
HOG	9	12	13	10	11	10	30
DPM	8	9	9	10	11	13	17
HYBRID	6	8	7	11	12	13	15

2. Frame Time, Sweep Time and Number of Glimpses

As previously discussed, the HOG method was the fastest computationally, followed by the BOW and then DPM methods. The computation time of the HYBRID method was dependent on the selected threshold values. The t_F for the HYBRID methods was measured using the same 100 positive and 100 negative, 1920 by 1080 evaluation images. Timing for the HYBRID method was measured in two ways, one when no positive detections were found and a t_F of 0.485 seconds was obtained. Then t_F was measured when a positive detection was obtained in the positive timing images. An example of the HYBRID positive detection is shown in Figure 23 for the positive timing evaluation image. In this timing evaluation there were four regions of interested selected by the HOG, three of which were not accepted by the BOW portion of the HYBRID detector. The evaluated t_F increased to 0.740 seconds when using the threshold values that produced a positive detection as seen in Figure 23. The rest of the evaluations were conducted with t_F of 0.740 seconds.

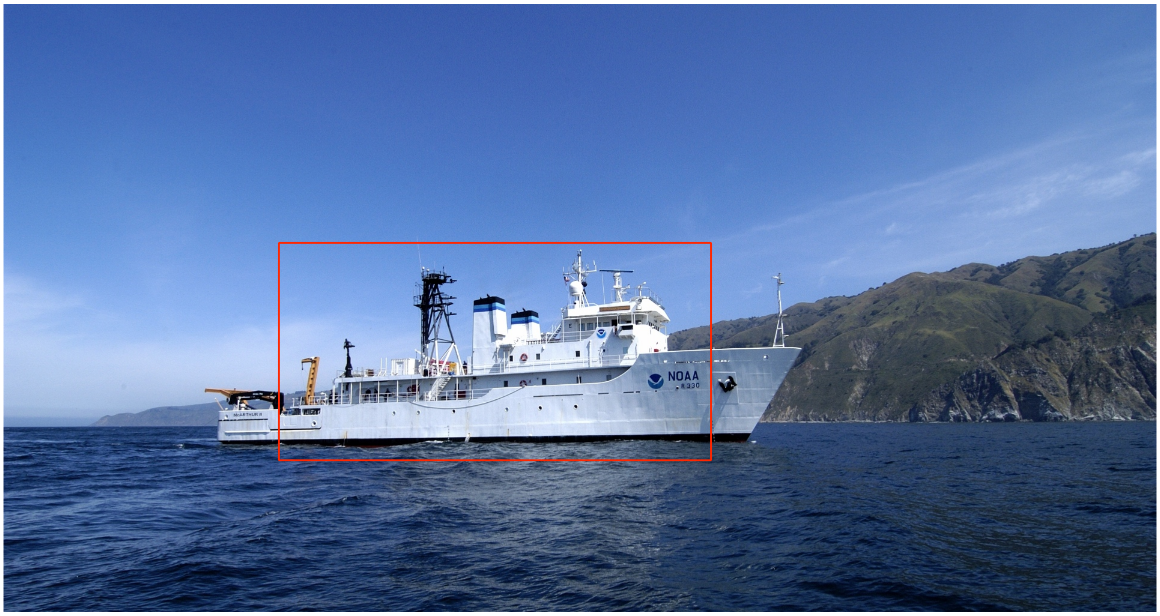

Figure 23. Example detection of the ship in the positive timing evaluation image by the HYBRID detector (after [39]).

The HYBRID method used the BOW implementation pre-scaling the already pre-scaled ROI by another factor of two; this implementation of BOW alone had a t_F of 1.88 seconds. The HOG detector of the HYBRID alone had a t_F of 0.485 seconds when timed with two-times pre-scaling. The HYBRID model met the minimum t_S for all FOV except UNFOV and was only fast enough to have multiple glimpses in the WFOV as shown in

Table 23 and Table 24. The ability to not have multiple glimpses made the calculated $P_D = \delta$ for calculating ω with the HYBRID and DPM detectors.

Table 23. The t_F and t_s of the top detector from each method by FOV.

Detector Method	t_F (seconds)	Sweep time t_s (seconds)			
		WFOV	MFOV	NFOV	UNFOV
BOW	0.199	30	30	60	60
HOG	0.117	30	30	60	60
DPM	2.0	30	49.5	132	330
HYBRID	0.740	30	30	60	122

Table 24. The number of glimpses of the top detector from each method by FOV.

Detector Method	t_F (seconds)	n (number of glimpses)			
		WFOV	MFOV	NFOV	UNFOV
BOW	0.199	15	6	5	2
HOG	0.117	25	11	8	3
DPM	2.0	1	1	1	1
HYBRID	0.740	4	1	1	1

3. Lateral Range Curves and Sweep Width

With $P_D = \delta$, the benefit of having a possible increase for detecting a ship through multiple glimpses is not available. The HYBRID t_F was fast enough to have more than 10 percent overlap per frame with the MFOV and NFOV lens and have multiple glimpses with the WFOV lens. Observed from every model was that the WFOV had such a low ω for all detector models that it would seem to be impractical as ship detection sensor for most operations. The MFOV lens had ω of nearly a decade larger than the WFOV but still may be impractical as a ship detection sensor. The NFOV and UNFOV lens obtained greater than 5 km ω for a 10-meter MHH contact, which may not be an acceptable range of detection for all operations. Even though the UNFOV could detect ships at much larger ranges than the NFOV, the NFOV also has some benefits. If the sensor were in rough seas, the horizon would be moving up and down in the image. The probability that the horizon could be maintained in every image capture increases for the

NFOV over the UNFOV. The HYBRID detector has similar ω as the other top detectors from each method when calculated using P_D as seen in Table 25.

Table 25. The ω of the top detector from each method calculated using $P_{D,}$ evaluated for a 10-meter MHH contact with a relative velocity of 20 knots.

Detector Model	Sweep Width ω (km)			
	WFOV	MFOV	NFOV	UNFOV
BOW	0.30	2.44	7.03	10.22
HOG	0.29	2.44	7.04	10.58
DPM	0.22	2.05	6.00	9.26
HYBRID	0.29	2.16	6.28	9.69

The HYBRID models ω was not as large as the upper bound ω of BOW and HOG detectors, shown in Table 25. The HYBRID model did surpass the DPM method in ω when calculated with P_D. Shown in Table 26 is the ω calculated based on δ for the top detectors from each model. The lateral range curves used to produce the ω in Table 26 are located in Appendix B, Lateral Range Curves, Figure 72 through Figure 75. The HYBRID model had the largest ω when calculated using δ for all FOVs. Of all the BOW models, the model that used GFTT and two-times pre-scaling had the largest ω when calculated using δ and was 9.27 km for UNFOV. The GFTT ω is still less than the HOG and HYBRID methods in Table 26.

Table 26. The ω of the top detector from each method calculated using δ, evaluated for a 10-meter MHH contact with a relative velocity of 20 knots.

Detector Model	Sweep Width ω (km)			
	WFOV	MFOV	NFOV	UNFOV
BOW (BRISK)	0.23	2.02	5.86	9.07
HOG	0.22	2.03	5.94	9.32
DPM	0.22	2.05	6.00	9.26
HYBRID	0.24	2.16	6.28	9.69

Provided in Figure 24 are the top detectors from each methods graph of ω by MHH for the UNFOV lens and a 20 knots relative velocity contact; this ω was calculated using δ. The HYBRID outperformed all other methods with this calculation of ω except for contacts less than one meter tall, in which HOG alone was superior. The DPM performed the worst on small contacts but then increased its performance above other methods as MHH increased. Also observed from this graph was that all of the detectors performed very well and were all very close in their performance. The differences in computational speed also impacted the ω when relative velocity was considered.

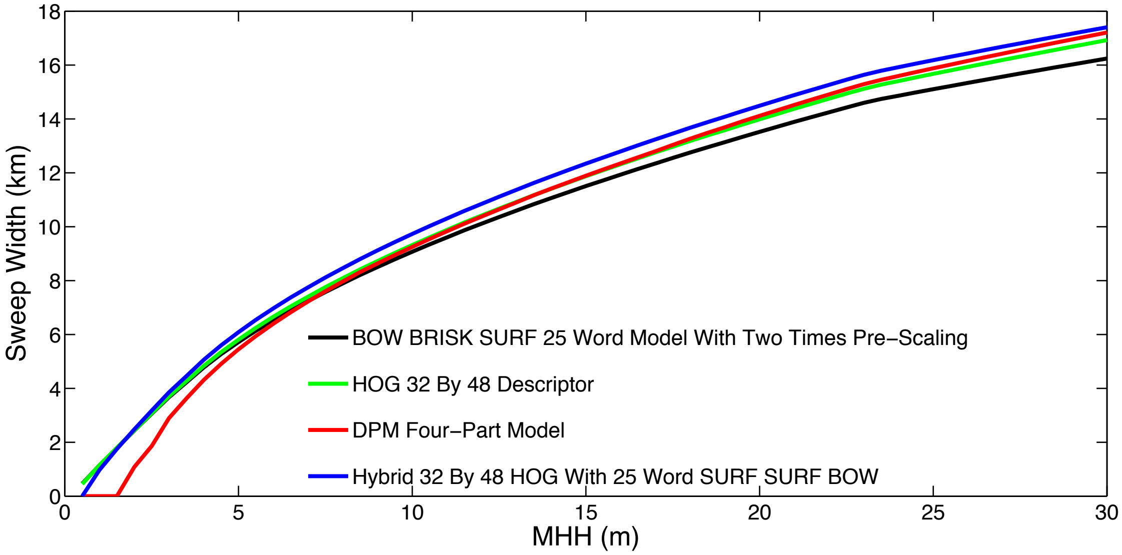

Figure 24. Graph of ω by MHH for the top detector of each method, based on UNFOV lens and a 20-knot relative velocity contact.

Provided in Figure 25 is the top detector from each method graph of ω by relative velocity for a 10-meter MHH contact. In this graph, as the contacts relative velocity increased, the distance that the contact traveled in-between detections was increased. The further the contact traveled, the closer the range of detection was when the lateral range was calculated. The distance that the contact could travel was entirely based on the t_S of the detector and relative velocity. The shorter t_S is, the shorter the distance traveled by the contact is between sweeps, and thus the closer the lateral range is to the range of the calculated δ. The detectors ω in Figure 25 is also calculated using δ and not P_D. The HYBRID method was only slightly superior in terms of ω for all relative velocities. The impact of the DPM methods slow computation time is shown as relative velocity is increased. The DPM method had the second largest ω for slow contacts and then had the lowest ω for the fastest contacts. All of these methods performed well in the evaluation model given that where these succeeded, thousands of trained ship detectors failed.

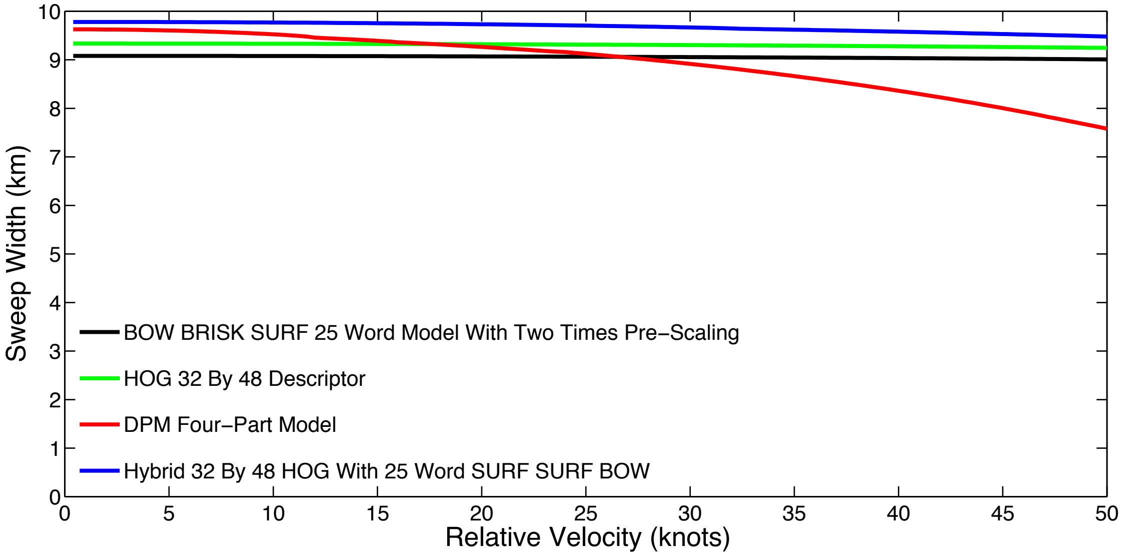

Figure 25. Graph of ω by relative velocity for the top detector of each method, based on UNFOV lens and a 10-meter MHH contact.

C. FUTURE RESEARCH

A design of experiments approach was not taken with the HYBRID detector, which was conducted with the other detection methods. There could still be room for improvement with adjusting the parameters that have already been built into the HYBRID detector. The scaling and number of level parameters along with the thresholding values used to produce the ROC curves can be adjusted dynamically and allow for optimization in a changing environment. The other BOW implementations may have also produced better results on the ROI produced by the HOG detector.

Instead of using a scale down and rescan approach for multiscale detection, creating HOG descriptors at each scale that better represent a ship at that scale could improve the HOG ship detection in speed of detection, recall and precision. Computational speed could be increased if this was performed on the GPU, allowing all of the descriptors to scan the image in parallel without requiring resizing the image. Recall and precision could be improved with the descriptors better representing the ships at every scale. Having different descriptors for different ship types may also improve recall and precision, as may having different descriptors for different aspects of ships, similar to the DPM poses.

Many of these methods were developed as multiclass detection methods, though the detectors were only trained as single class (ships) or two classes (ships or non-ships) detectors. The BOW detector in this research was programed as a multiclass detector and could be trained as such. The detectors could return better recall results by more useful information, such as identifying a war ship or a merchant ship. The DPM by iRobot was implemented on a GPU and is also multiclass, trainable to distinguish between vessel types, though the computation time increased for each additional class [47]. The HOG detector by itself is not usually used for multiclass detection, but the HYBRID approach could use the multiclass capabilities of BOW to determine class of vessel. Until a computer can, in real time, detect and classify ships with 100 percent recall and precision, there will continue to be future research and room for improvement.

There was also room for improvement in the evaluation model. The model was designed to represent a photonics mast sensor on a submarine, though this model could be extended to many platforms. The model also has room for improvement based on the many assumptions in making the model. The assumption that the detectors could only detect a vessel if more than half of the ship was above the horizon placed a hard limit on the detection range for developing the lateral range curves and, subsequently, for calculating ω. This visual detection limit may have been conservative or may have been generous; fine-tuning the model requires further research. Future research could also develop a more accurate increase in detection probability for multiple glimpses with a visual sensor.

Limits were also put on how fast the sensor could turn and accurately acquire image frames. Having actual hardware specification for the rotating equipment and light sensitivity of the sensor could help to improve the evaluation model. Changing the sensor and lens to meet a detection range and probability specification could also optimize the detection methods evaluated. When considering the implementation of the evaluation model for determining the operational performance of ship detectors, it may be important to base the calculation of δ off a maximum allowable false positive probability instead of the point closest to 100 percent recall and precision. The evaluation model was not limited by a set false positive rate.

V. CONCLUSIONS AND RECOMMENDATIONS

In search of a computer vision algorithm that could replace or aid operators in open seas and harbor ship detection, an evaluation model was created based on search theory evaluation methods for SONAR and RADAR systems. Using a larger focal length and narrower fields of view provided the greatest increase in detection range for all ship detection methods evaluated. The restriction and assumption of the evaluation method limited the computer vision algorithms to a maximum sweep width ω, or detection range, of 10.88 km for a 10-meter MHH contact traveling with a relative velocity of 20 knots. The optimized BOW, HOG and DPM object detection methods as open sea and harbor ship detectors all had ω of over 9.0 km for this same 10-meter MHH contact.

The HYBRID ship detector had a 9.69 km ω for the same 10-meter MHH contact. The HYBRID detected ships larger than 125 pixels tall with greater than 93 percent recall and a false positive rate of 8 percent or less. The HYBRID detector also detected ships between 38 and 64 pixels tall with an 86 percent to 92 percent recall and 11 percent to 13 percent false positive rate. This HYBRID detector had the lowest false positive rate, at 15 percent, of all the evaluated ship detectors on the smallest ship images tested. The smallest ships tested had a mean pixel height of 25 pixels tall, and the recall for these small ships was 76 percent for the HYBRID model. Detecting small ships in imagery was found to be the most difficult challenge for all the ship detection methods. The evaluation model purposely challenged the computer vision algorithms by having negative evaluation images that contained many objects that could cause false positive detections. The HYBRID detector was the most successful at overcoming these challenges.

The HYBRID detector provides many qualities desirable for an operational ship detector for use with a submarine photonics mast. Firstly, the HYBRID detector has the benefits of multiscale ship detection done on a GPU, being able to detect ships of many sizes or at many ranges from the sensor very quickly using parallel processing. Additionally, the HYBRID detector localizes the ship detection in the image, which provides better bearing accuracy to a fire control system than a classification method such as BOW alone. Finally, the additional BOW stage of the HYBRID detector provides the increase in recall rate for ships that appear small in imagery while also reducing the false positive rate for all ship sizes.

A recommendation to improve the HYBRID model is to utilize the multiclass capabilities of the BOW algorithm to further categorize detected ships into classes. Another recommendation that could greatly improve the HYBRID method is to create multiple scales of descriptors for multiple ship aspects that can be parsed over the image in parallel on a GPU. The multiple scales of descriptors would increase the detection probabilities at the different scales, as shown from having descriptors of different sizes in the HOG evaluation. The ability to process the multiple descriptors in parallel on the GPU, without having to resize and rescan the image, should also improve computational speed. Operational testing and comparison of these algorithms would be required to consider these computer vision detectors to replace an operator for ship detection from a photonics mast. The high detection rates and low false positive rates along with the speed of computation makes the HYBRID method a good candidate to aid operators as an automatic visual ship detector.

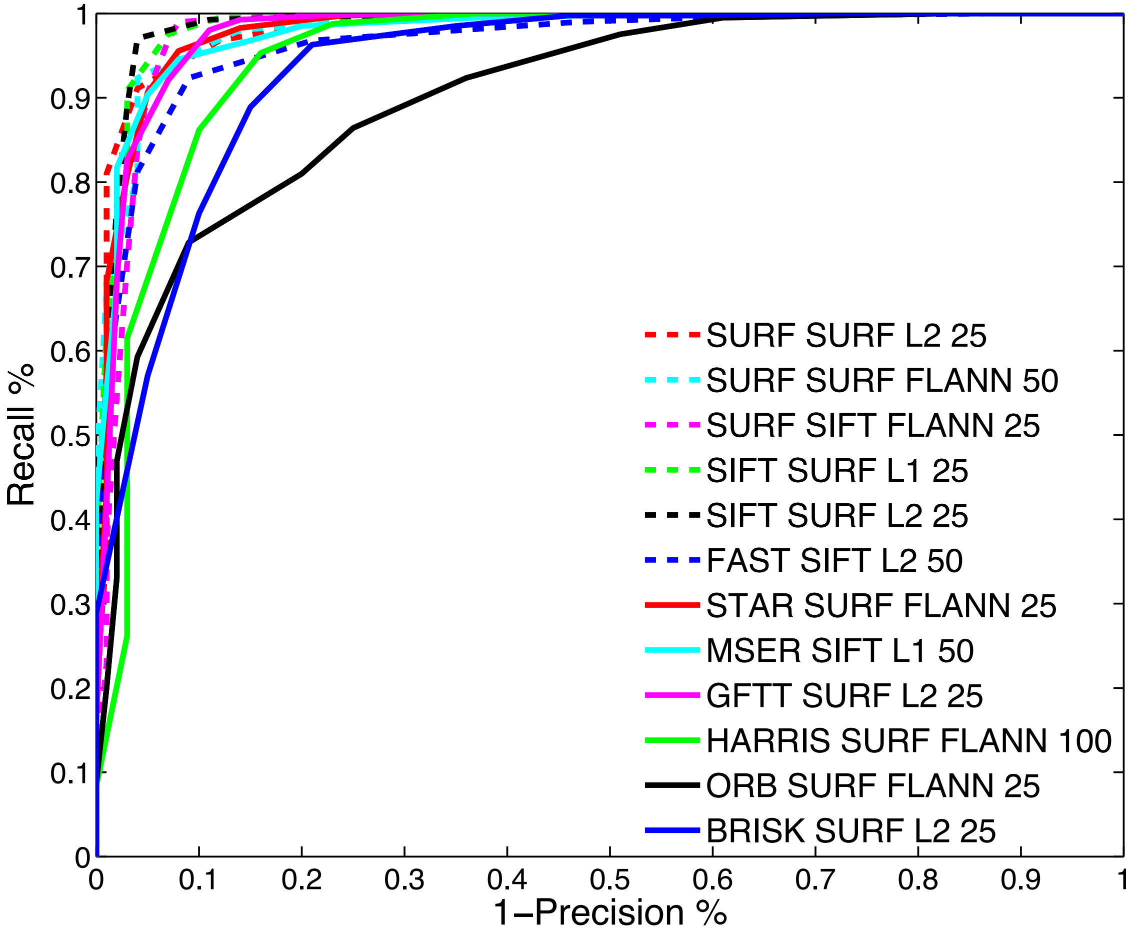

Figure 26. ROC curves for the 12 best BOW detectors on full-scale images.

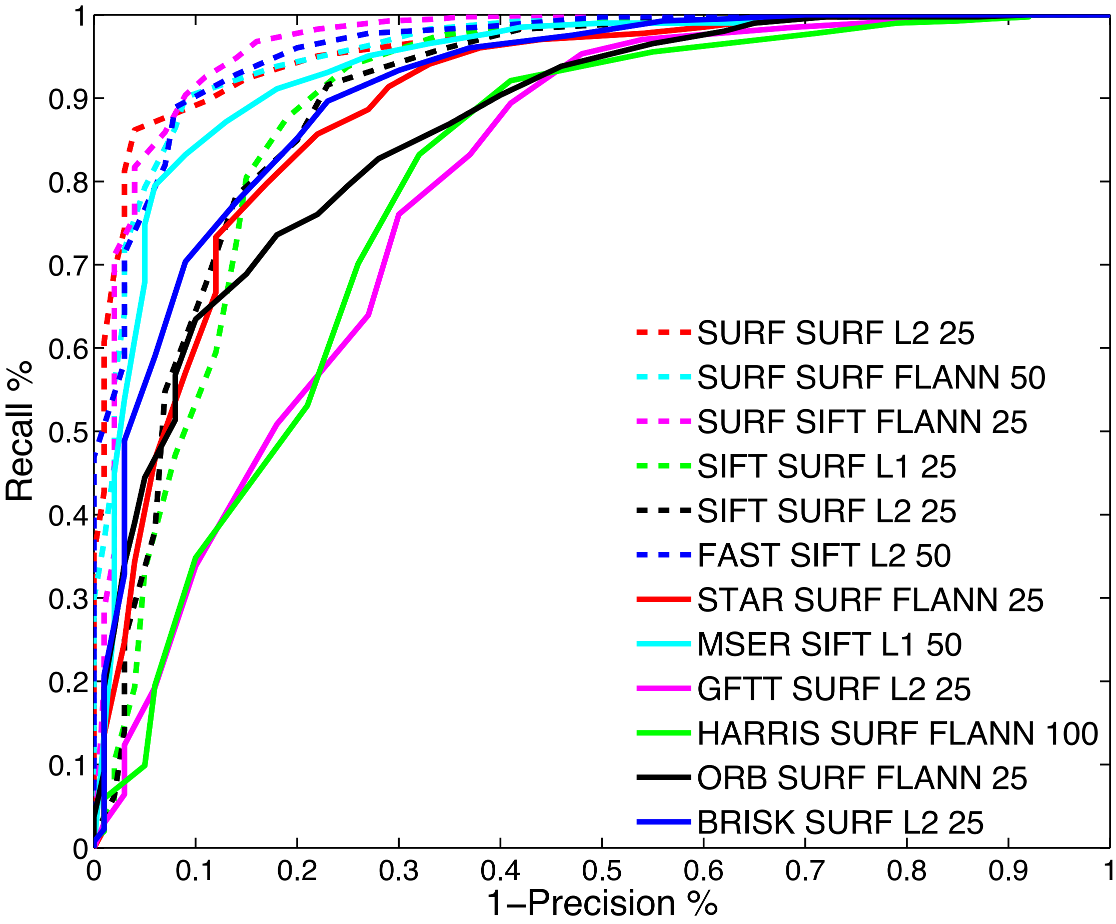

Figure 27. ROC curves for the 12 best BOW detectors on 75 percent scale images.

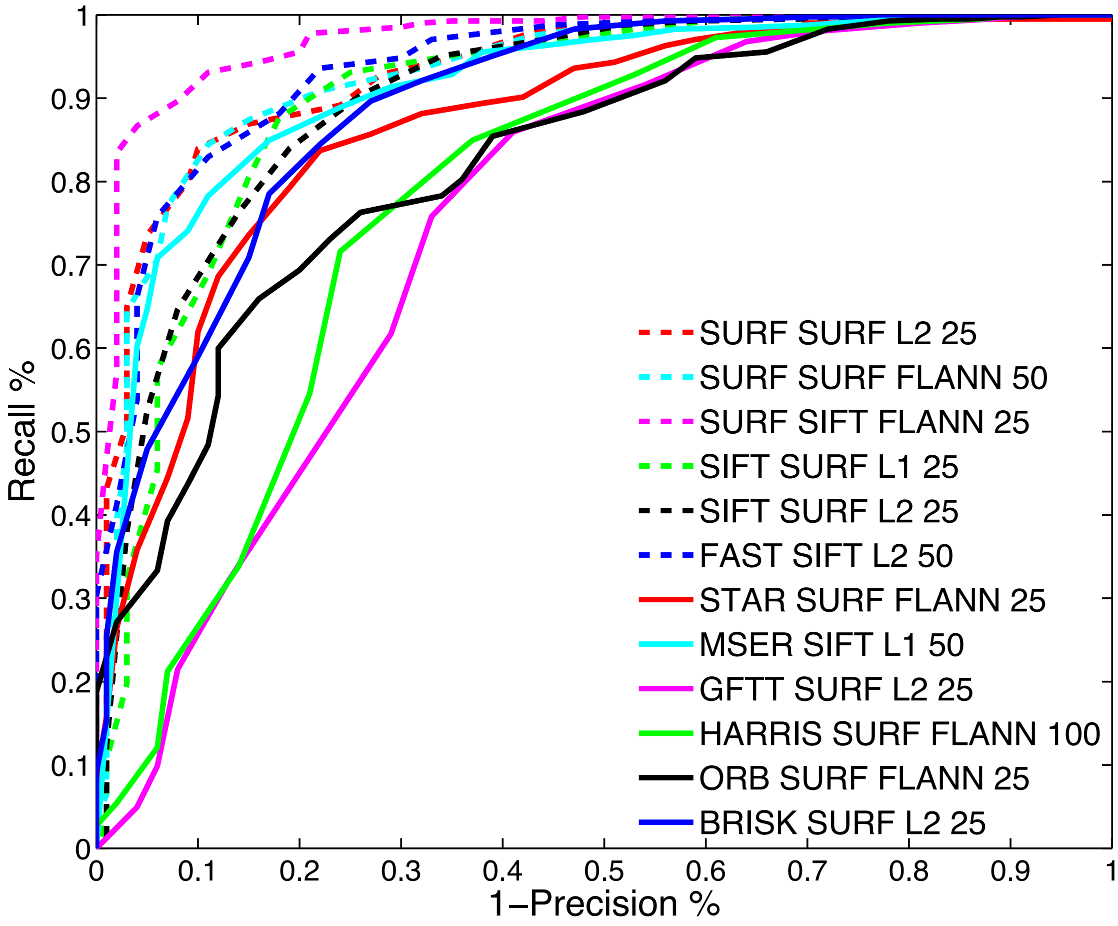

Figure 28. ROC curves for the 12 best BOW detectors on 50 percent scale images.

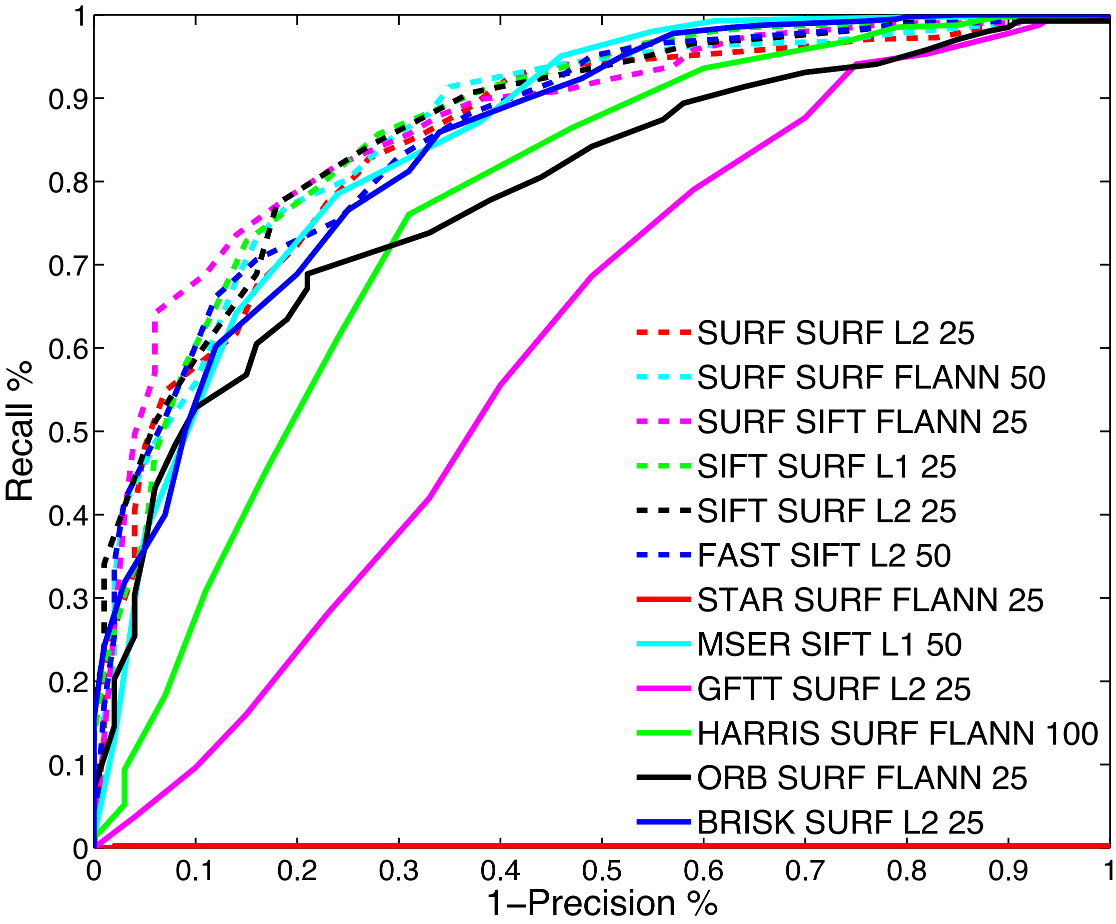

Figure 29. ROC curves for the 12 best BOW detectors on 25 percent scale images.

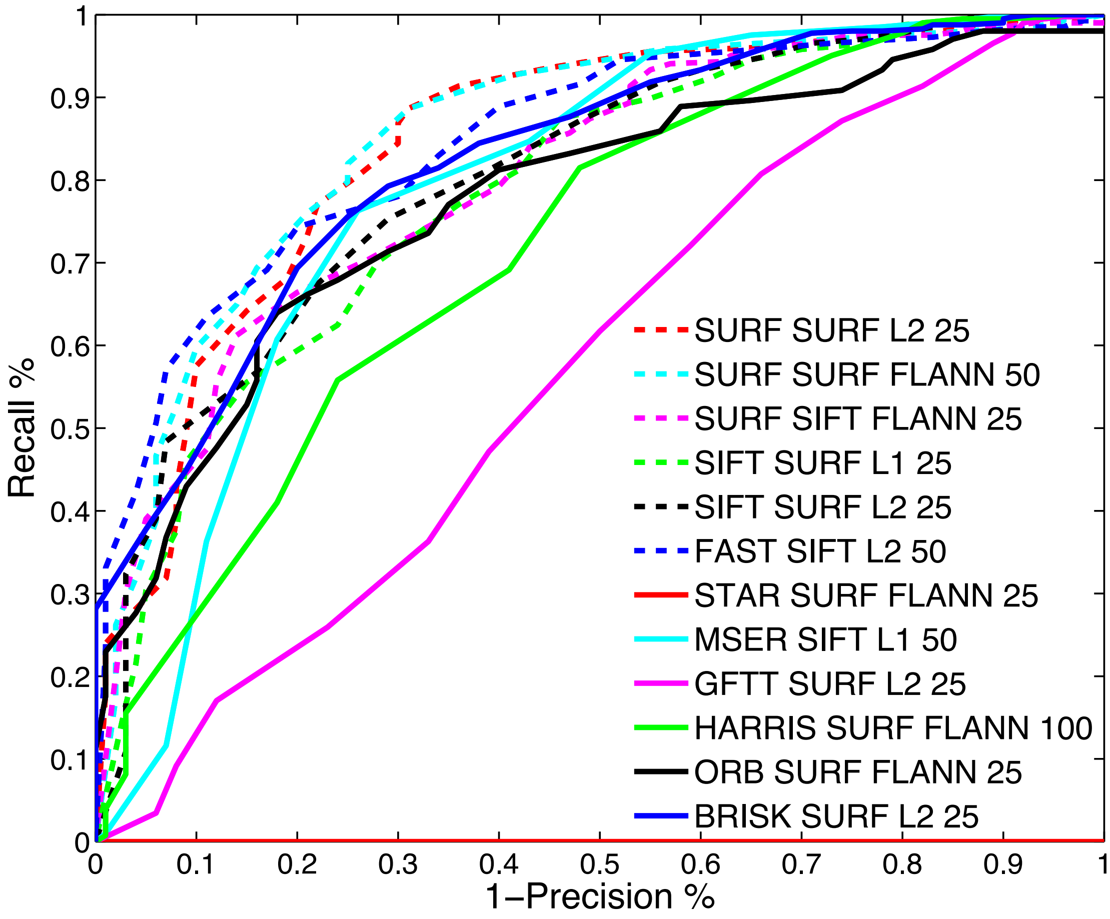

Figure 30. ROC curves for the 12 best BOW detectors on 20 percent scale images.

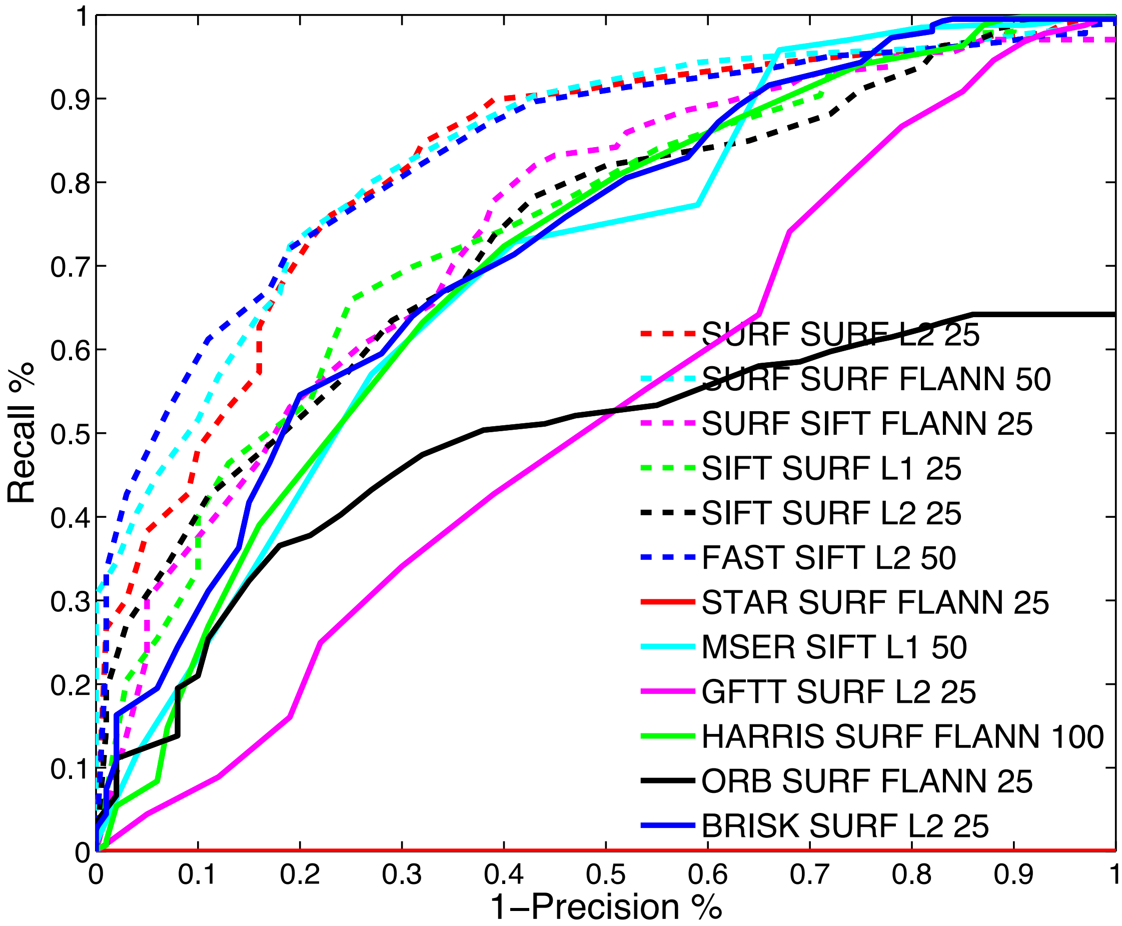

Figure 31. ROC curves for the 12 best BOW detectors on 15 percent scale images.

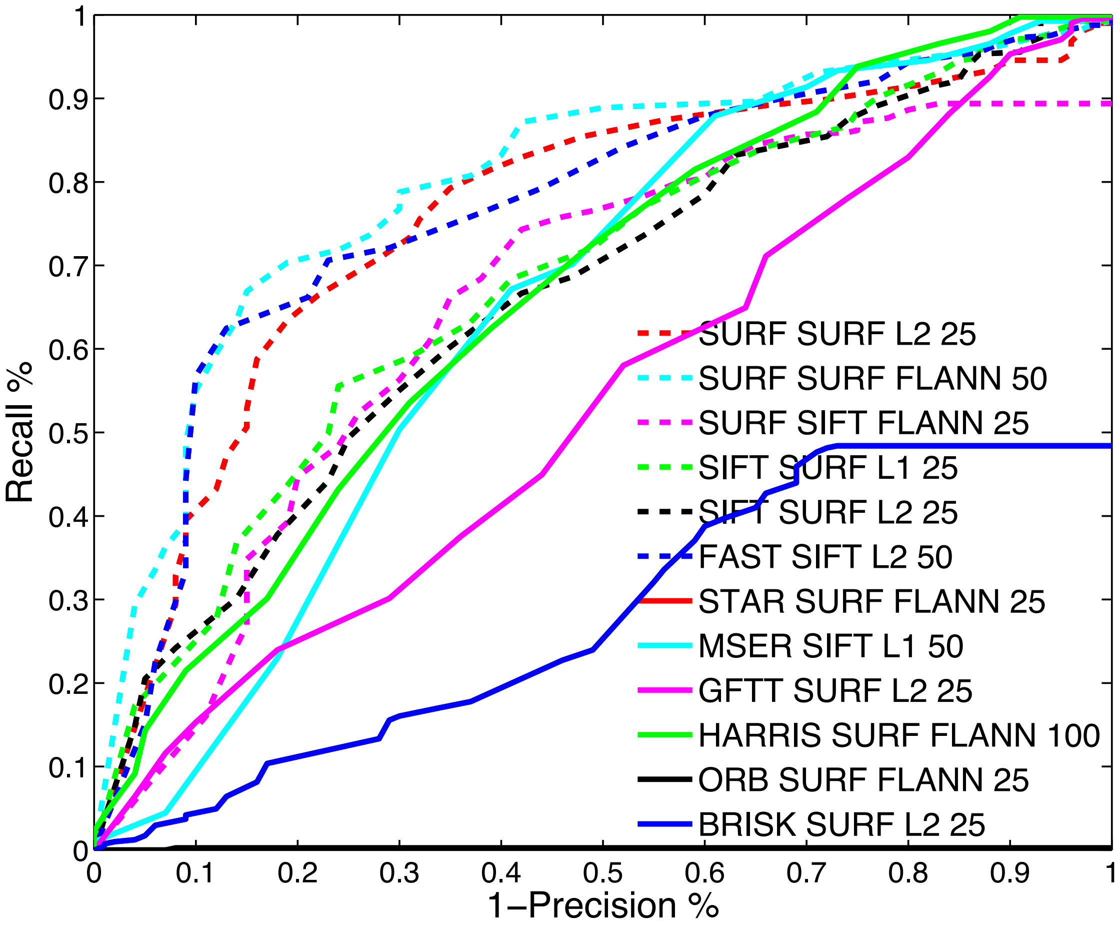

Figure 32. ROC curves for the 12 best BOW detectors on 10 percent scale images.

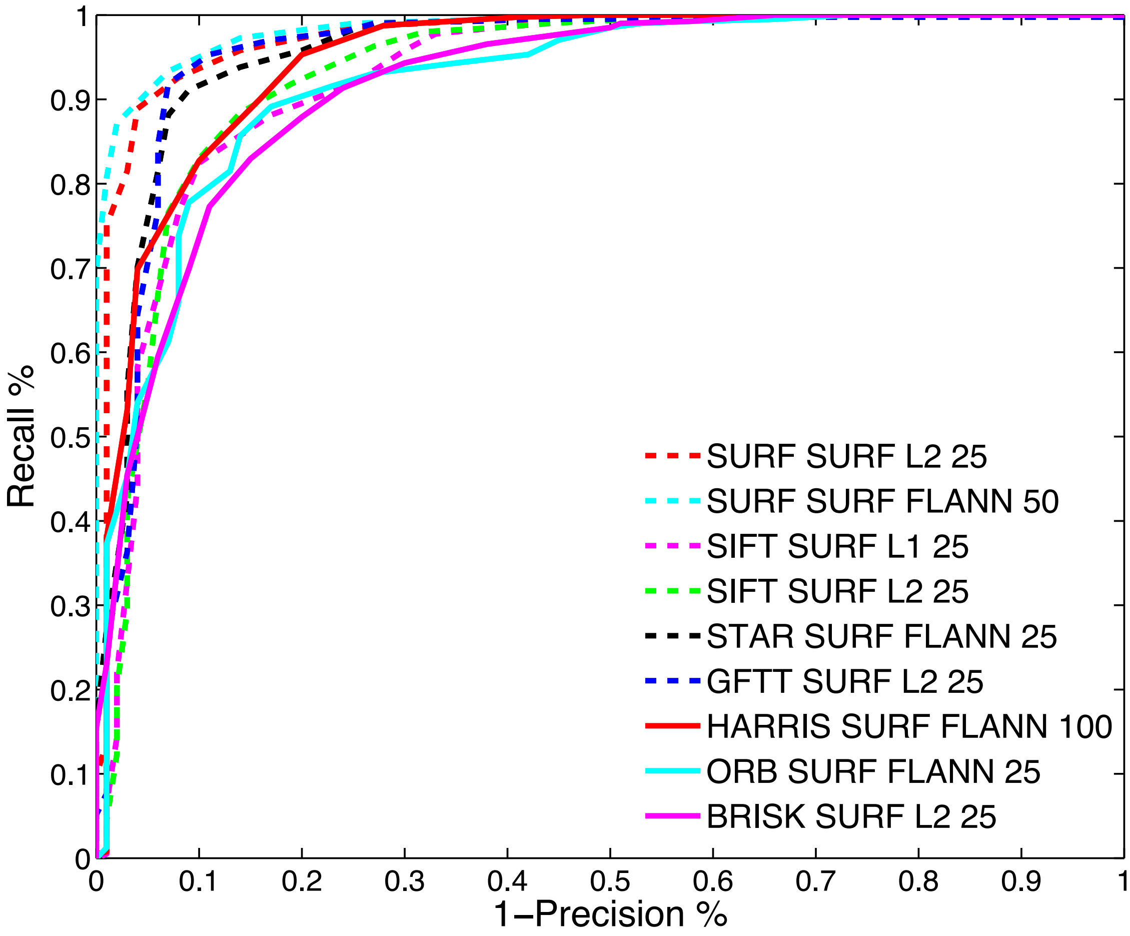

Figure 33. ROC curves for the nine fastest BOW detectors on full-scale images pre-scaling with a factor of two.

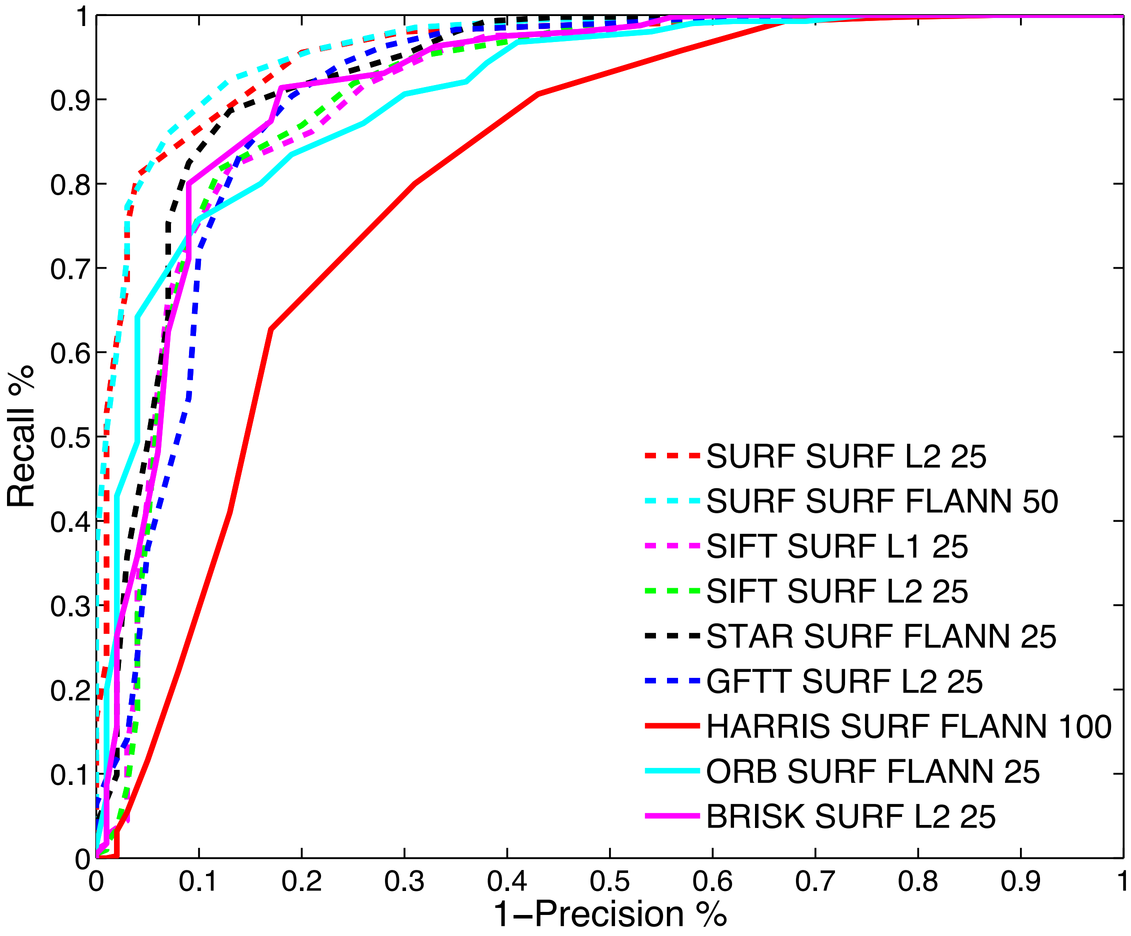

Figure 34. ROC curves for the nine fastest BOW detectors on 75 percent scale images pre-scaling with a factor of two.

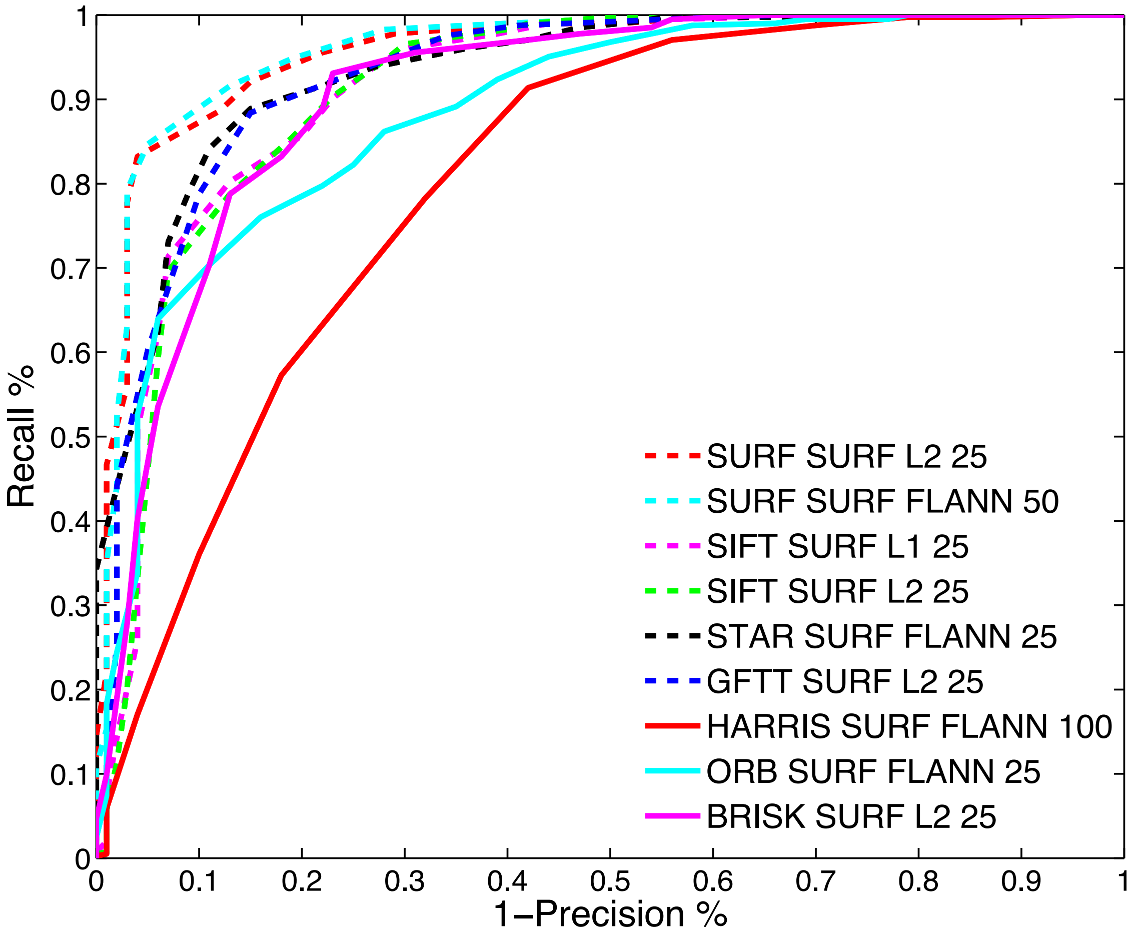

Figure 35. ROC curves for the nine fastest BOW detectors on 50 percent scale images pre-scaling with a factor of two.

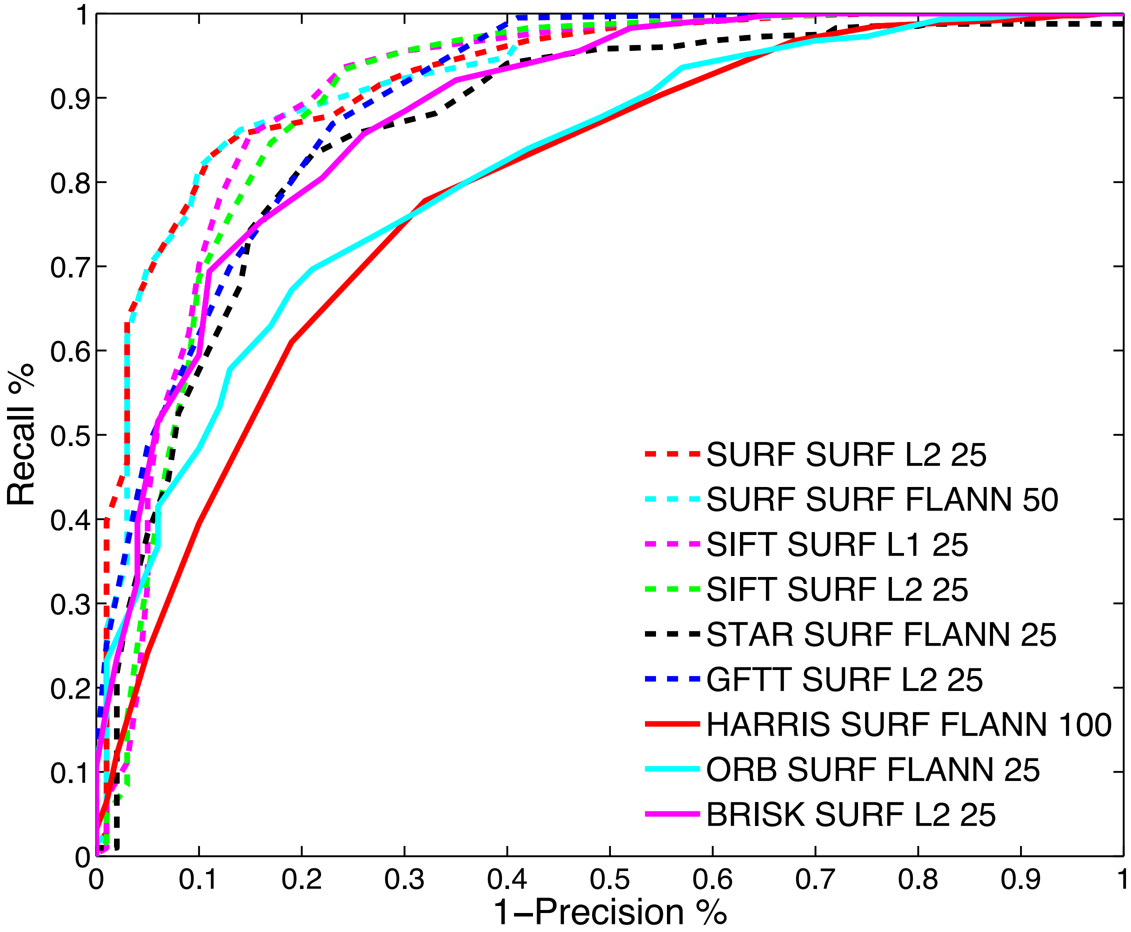

Figure 36. ROC curves for the nine fastest BOW detectors on 25 percent scale images pre-scaling with a factor of two.

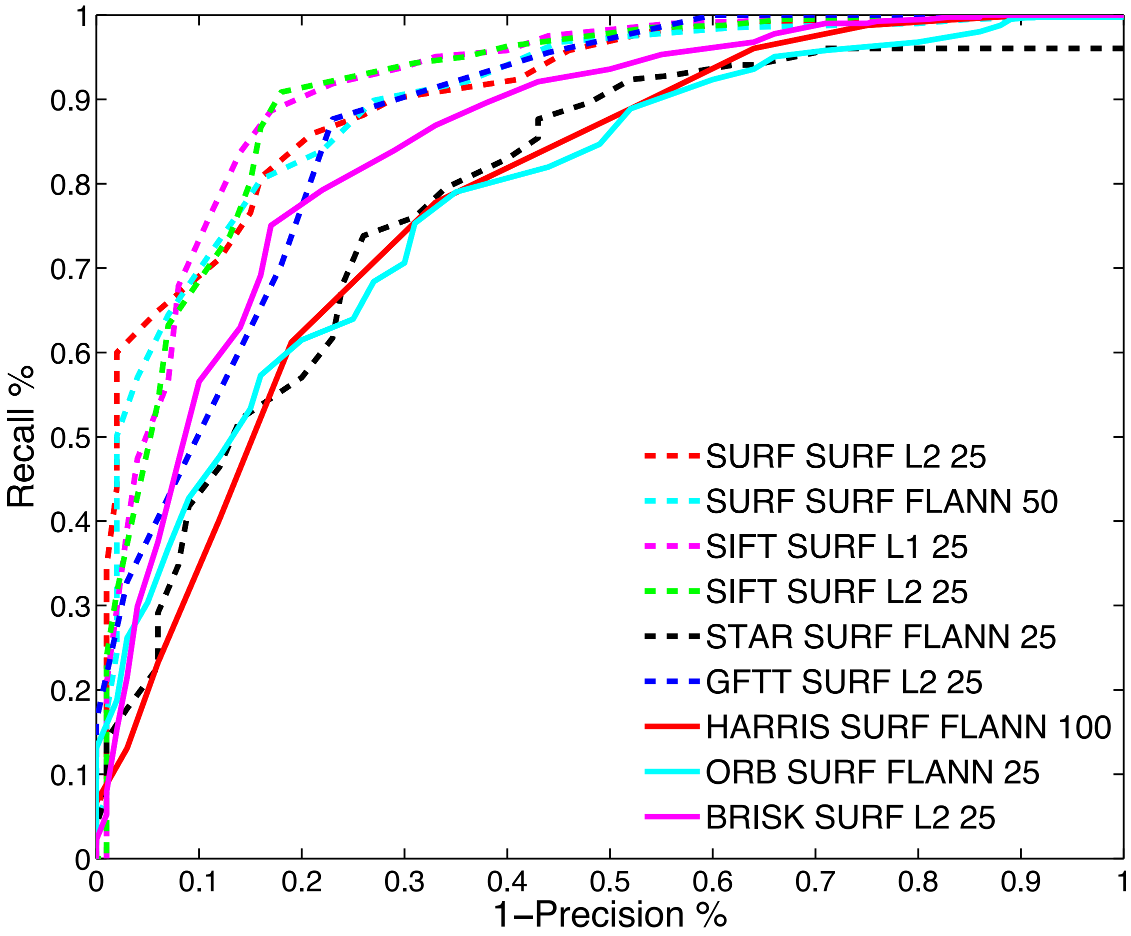

Figure 37. ROC curves for the nine fastest BOW detectors on 20 percent scale images pre-scaling with a factor of two.

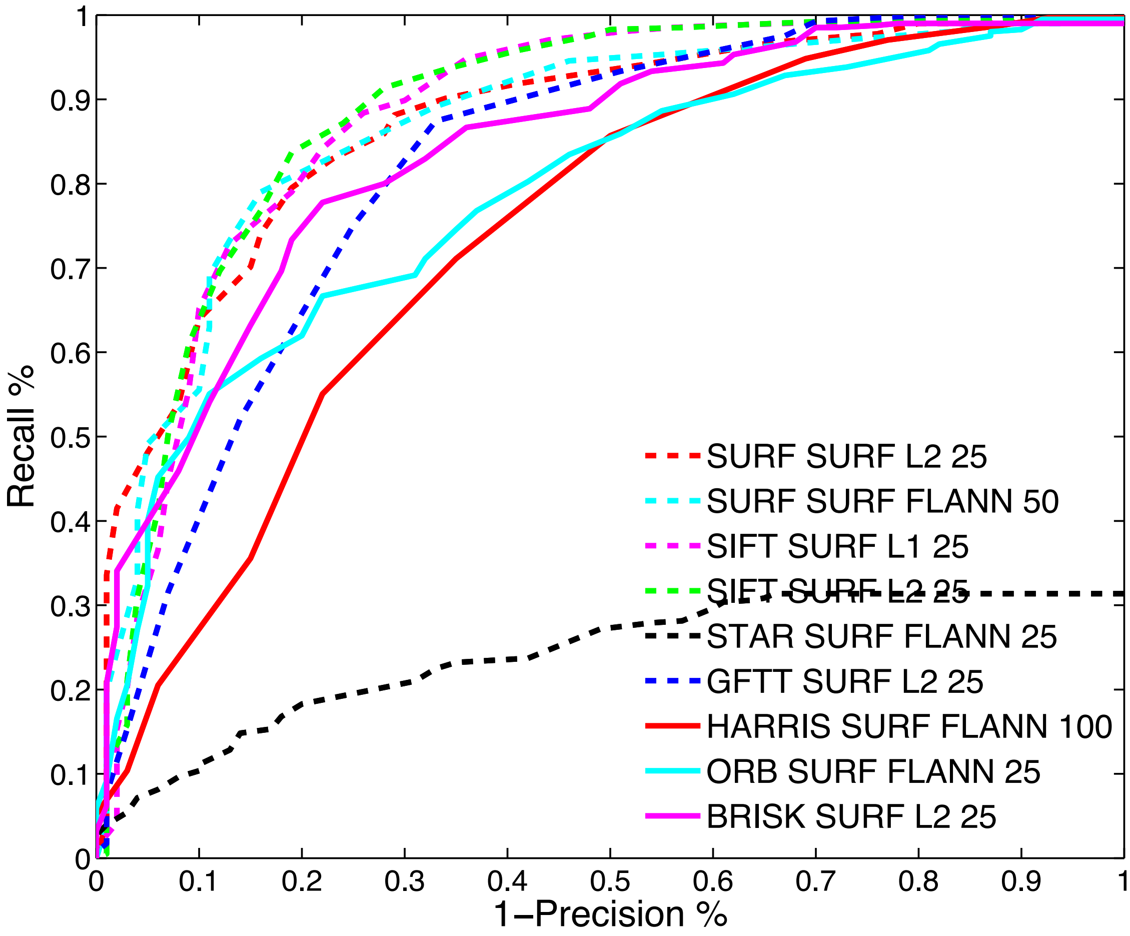

Figure 38. ROC curves for the nine fastest BOW detectors on 15 percent scale images pre-scaling with a factor of two.

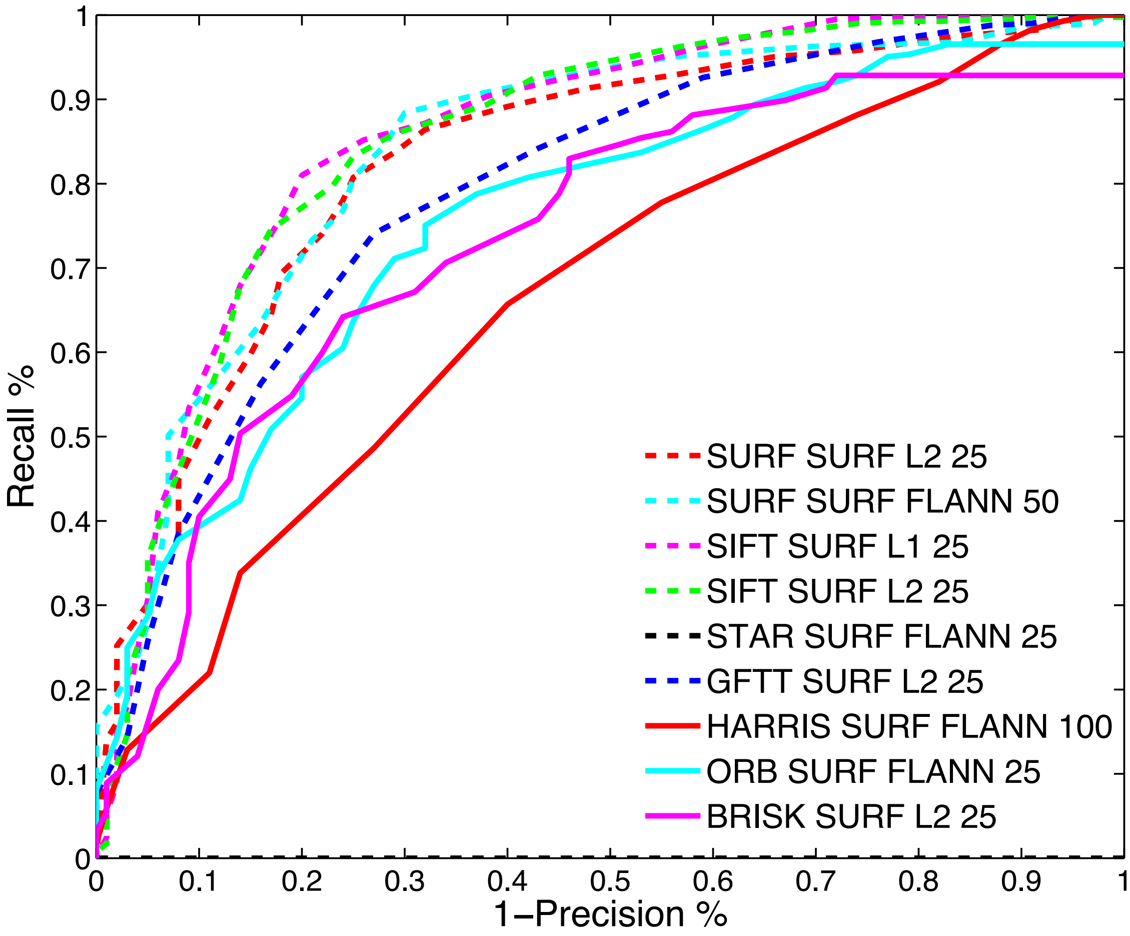

Figure 39. ROC curves for the nine fastest BOW detectors on 10 percent scale images pre-scaling with a factor of two.

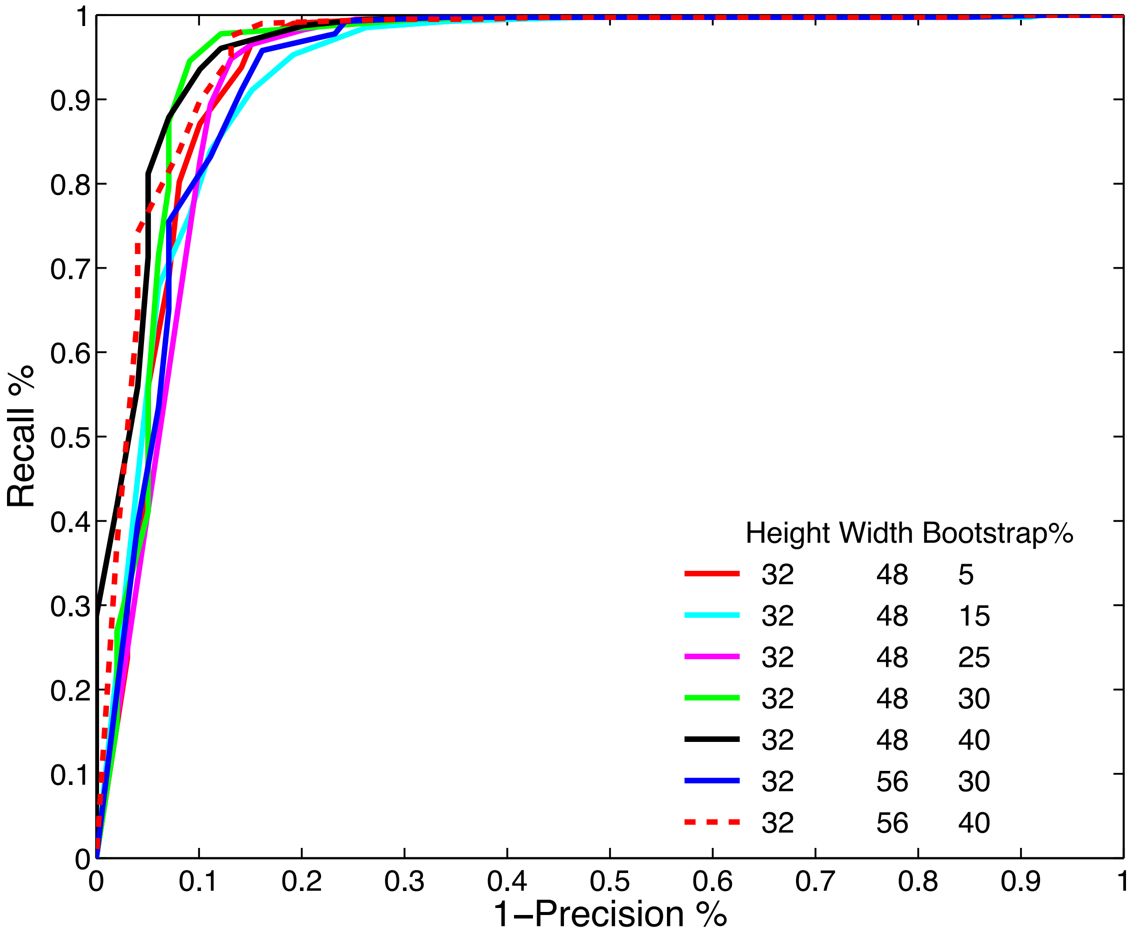

Figure 40. ROC curves for the top seven HOG detectors on full-scale images preformed on CPU.

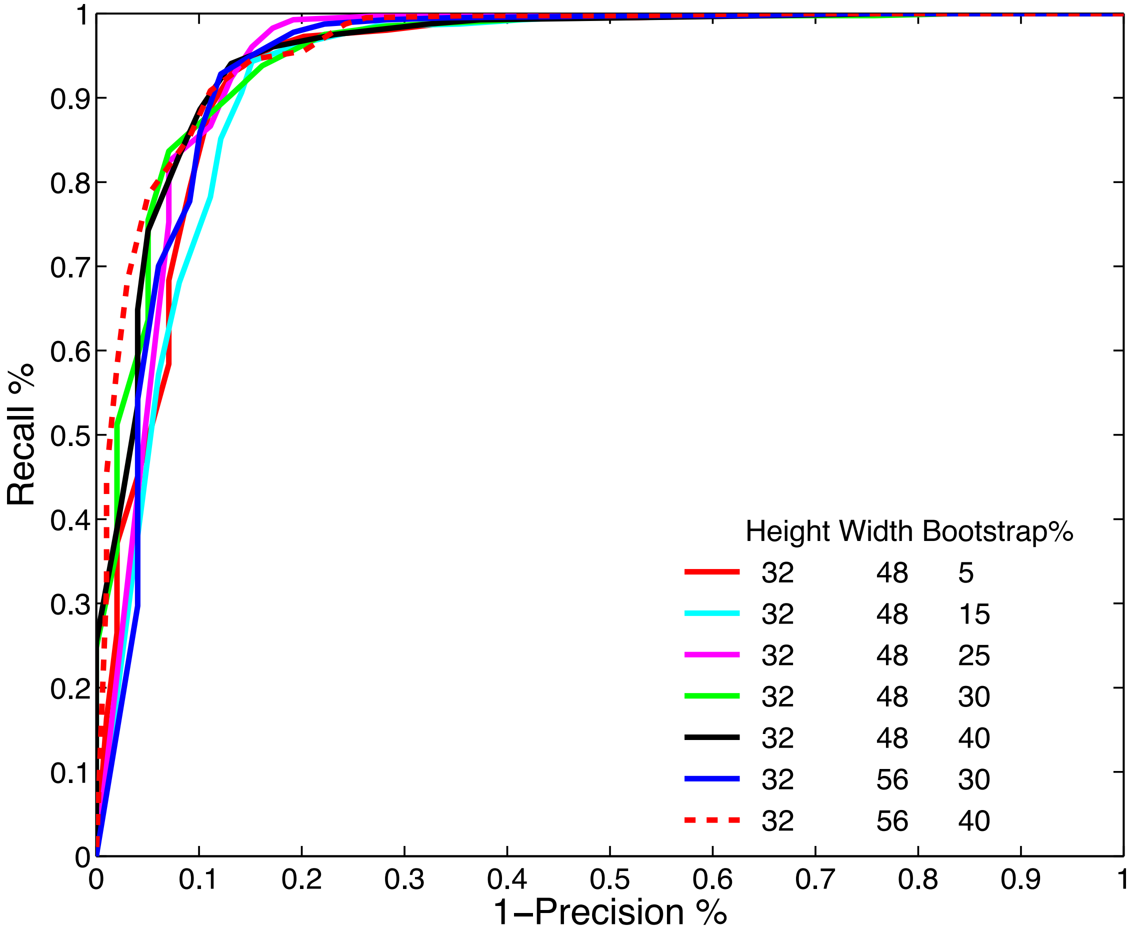

Figure 41. ROC curves for the top seven HOG detectors on 75 percent scale images preformed on CPU.

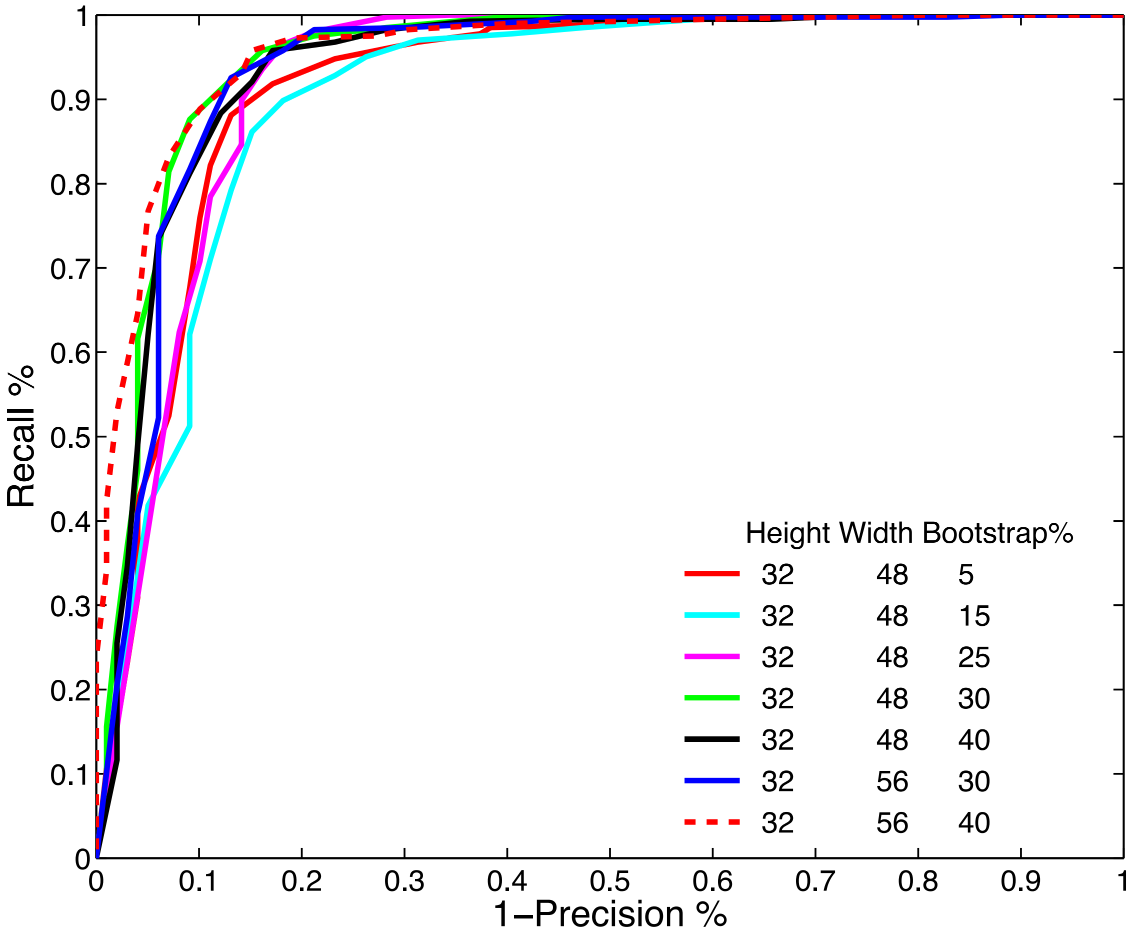

Figure 42. ROC curves for the top seven HOG detectors on 50 percent scale images preformed on CPU.

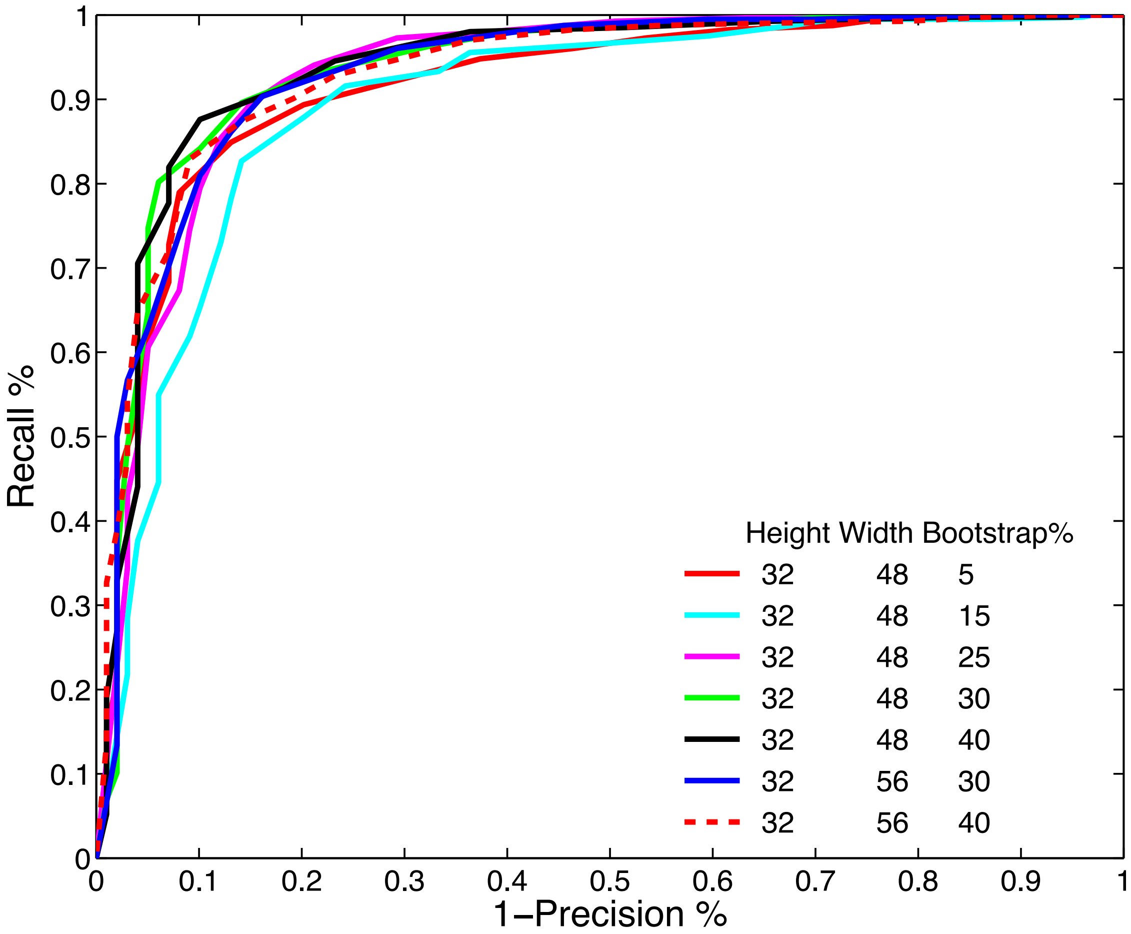

Figure 43. ROC curves for the top seven HOG detectors on 25 percent scale images preformed on CPU.

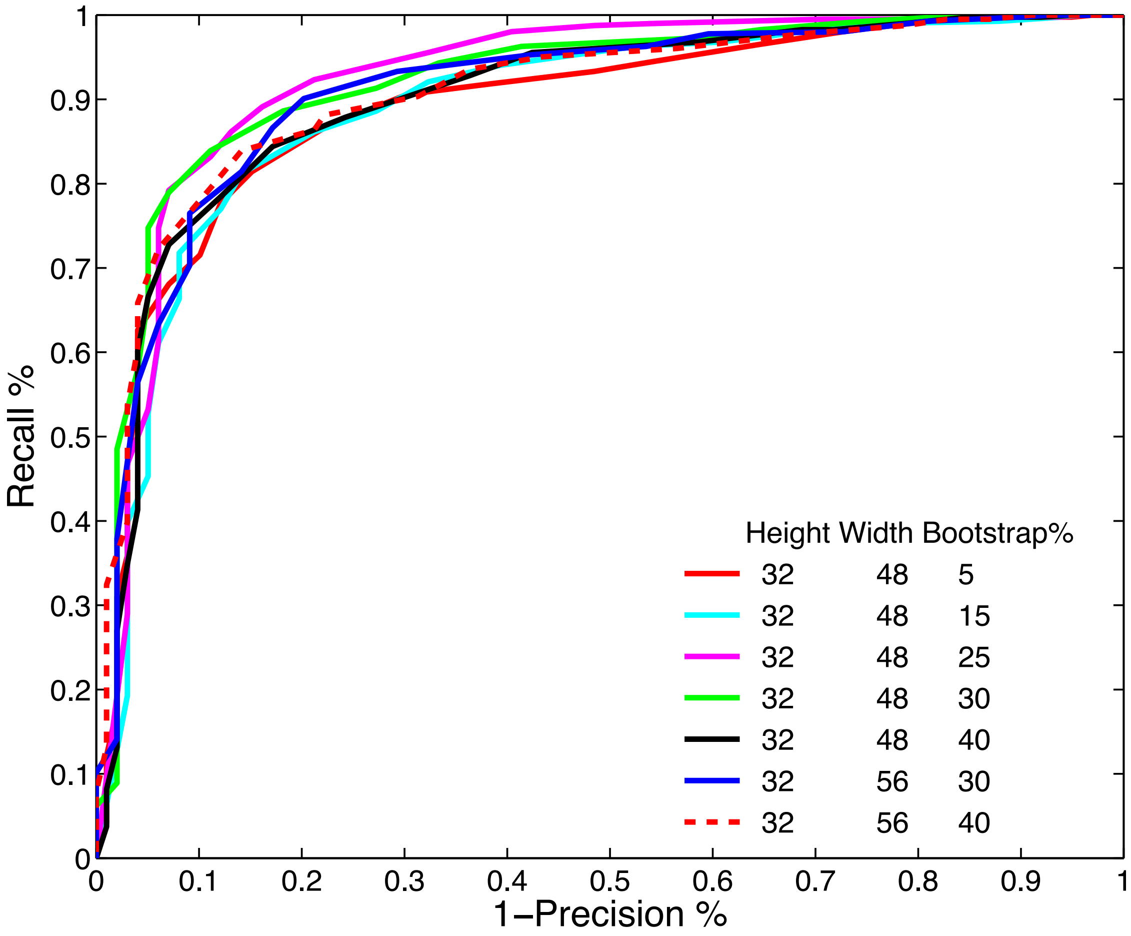

Figure 44. ROC curves for the top seven HOG detectors on 20 percent scale images preformed on CPU.

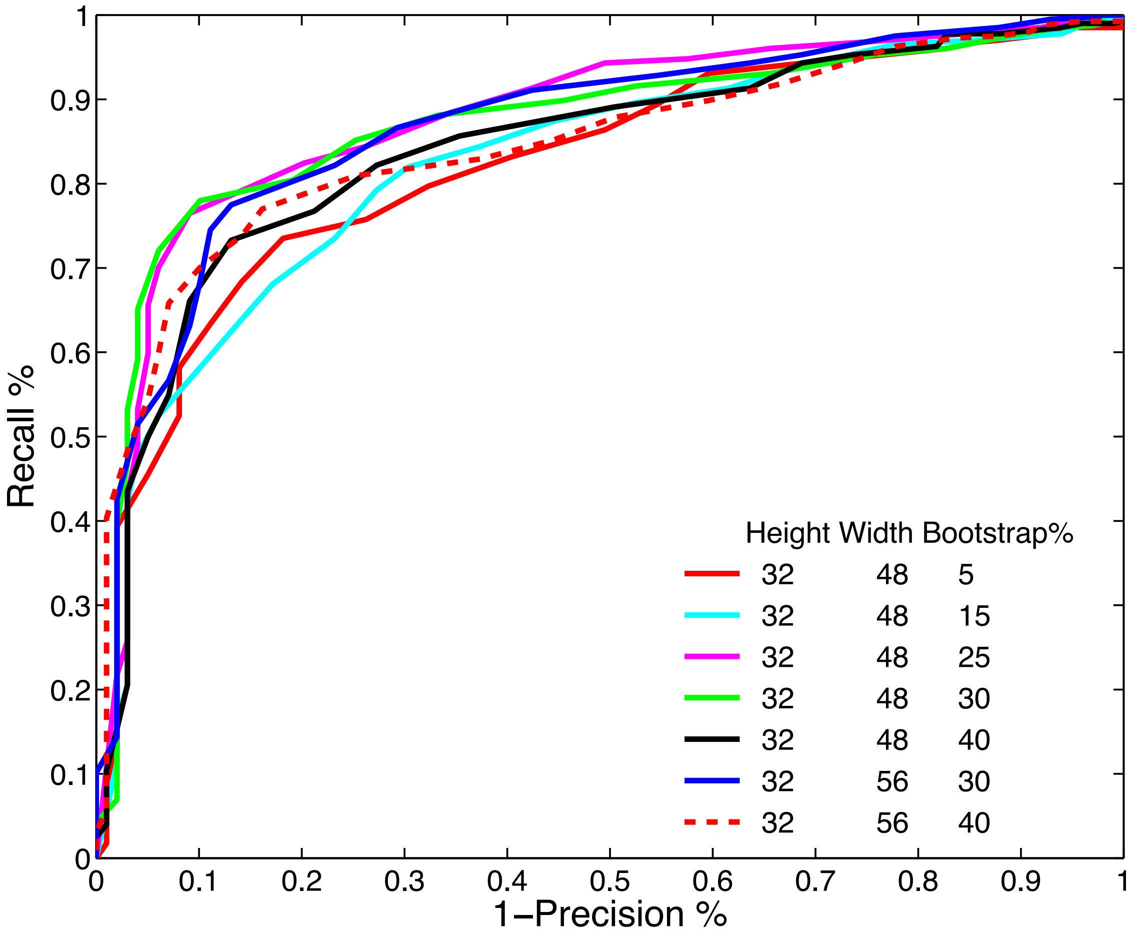

Figure 45. ROC curves for the top seven HOG detectors on 15 percent scale images preformed on CPU.

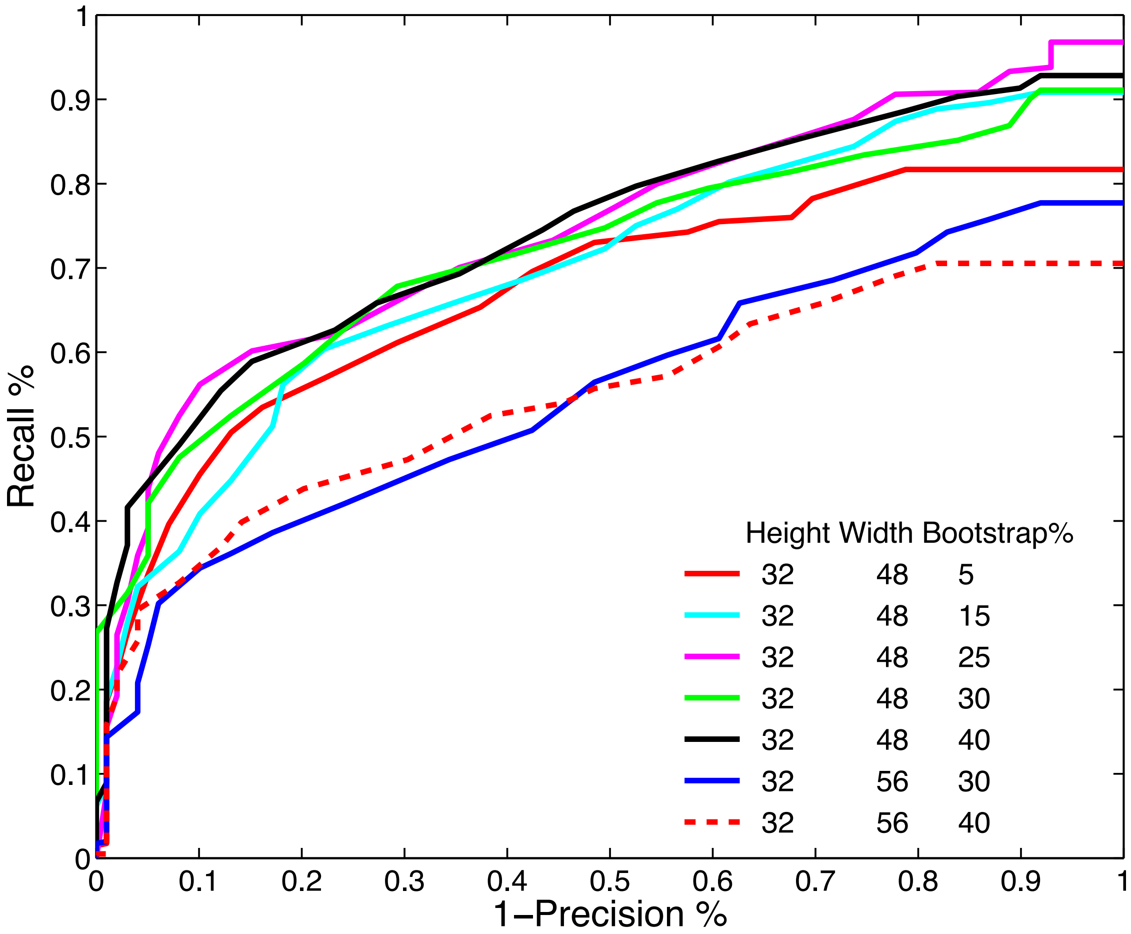

Figure 46. ROC curves for the top seven HOG detectors on 10 percent scale images preformed on CPU.

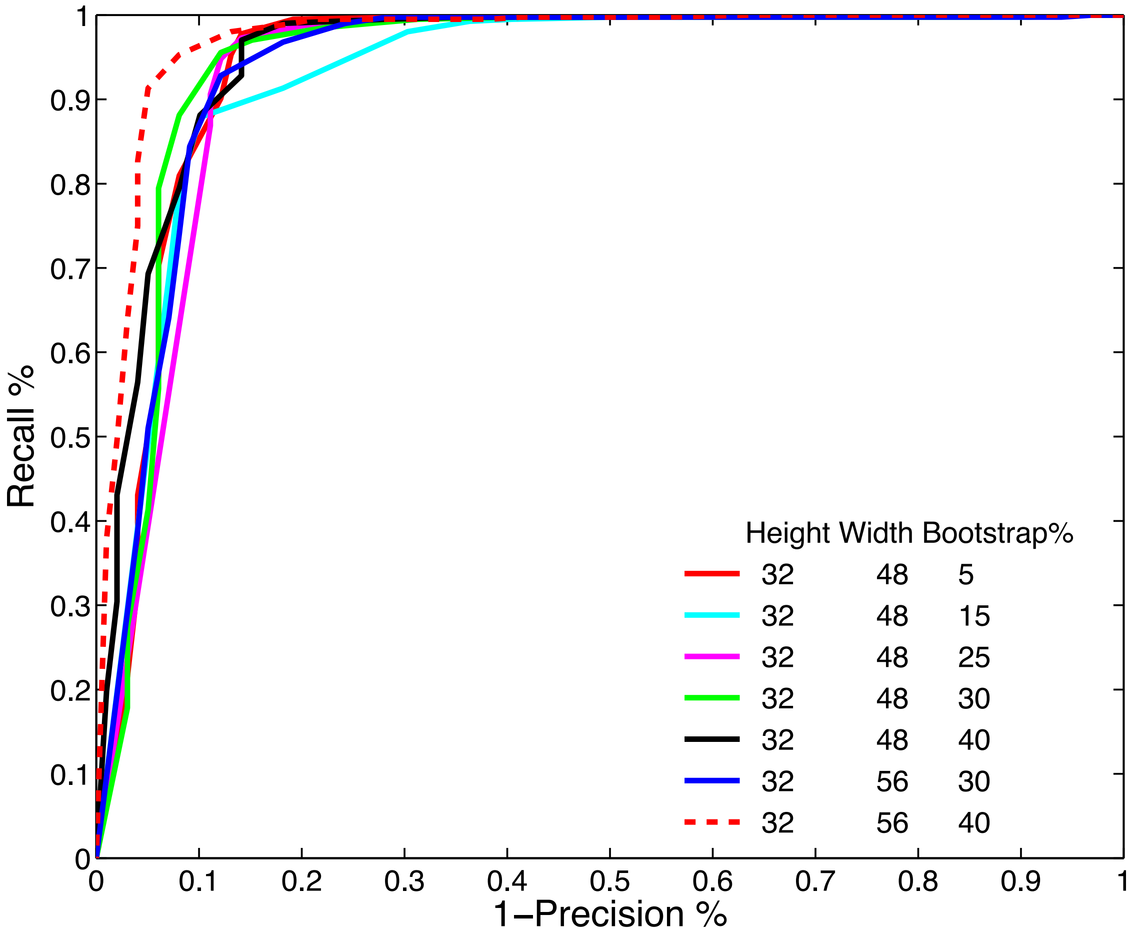

Figure 47. ROC curves for the top seven HOG detectors on full-scale images preformed on GPU.

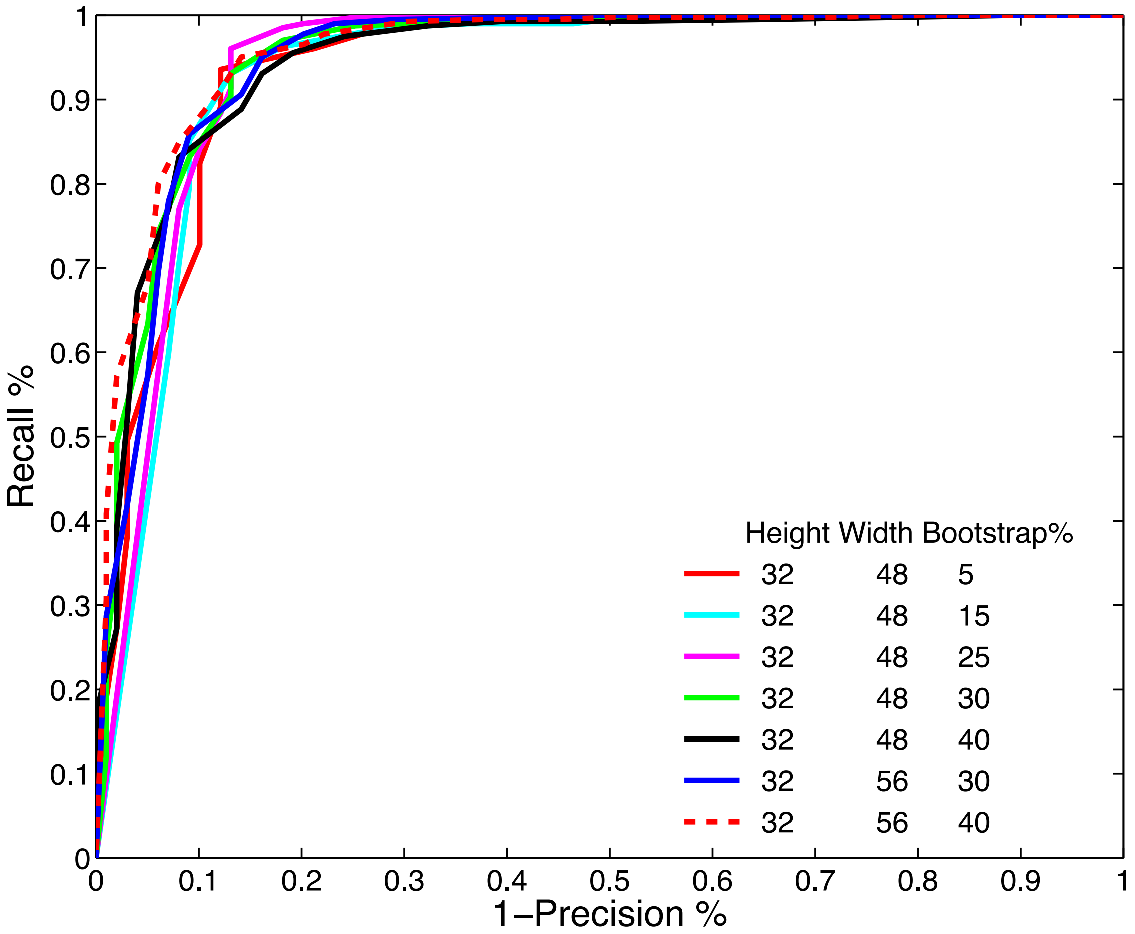

Figure 48. ROC curves for the top seven HOG detectors on 75 percent scale images preformed on GPU.

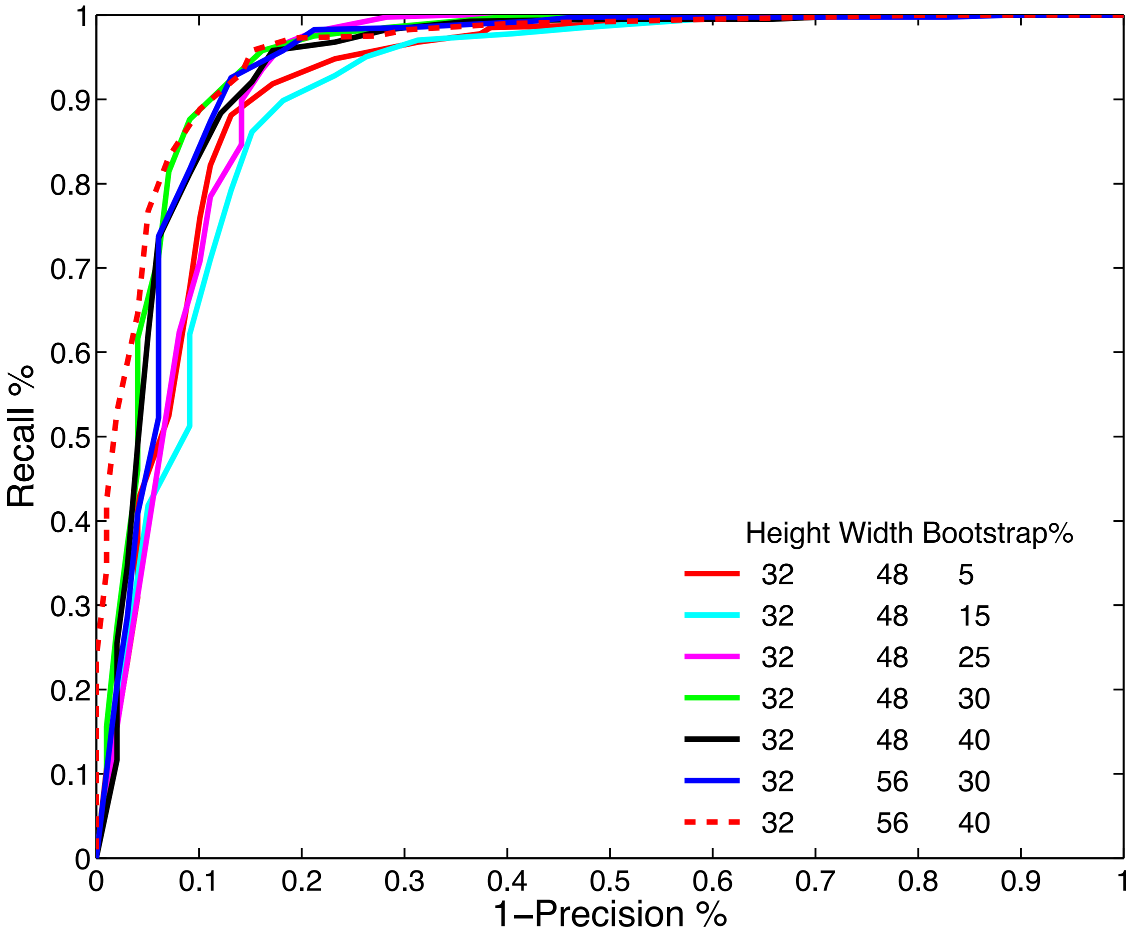

Figure 49. ROC curves for the top seven HOG detectors on 50 percent scale images preformed on GPU.

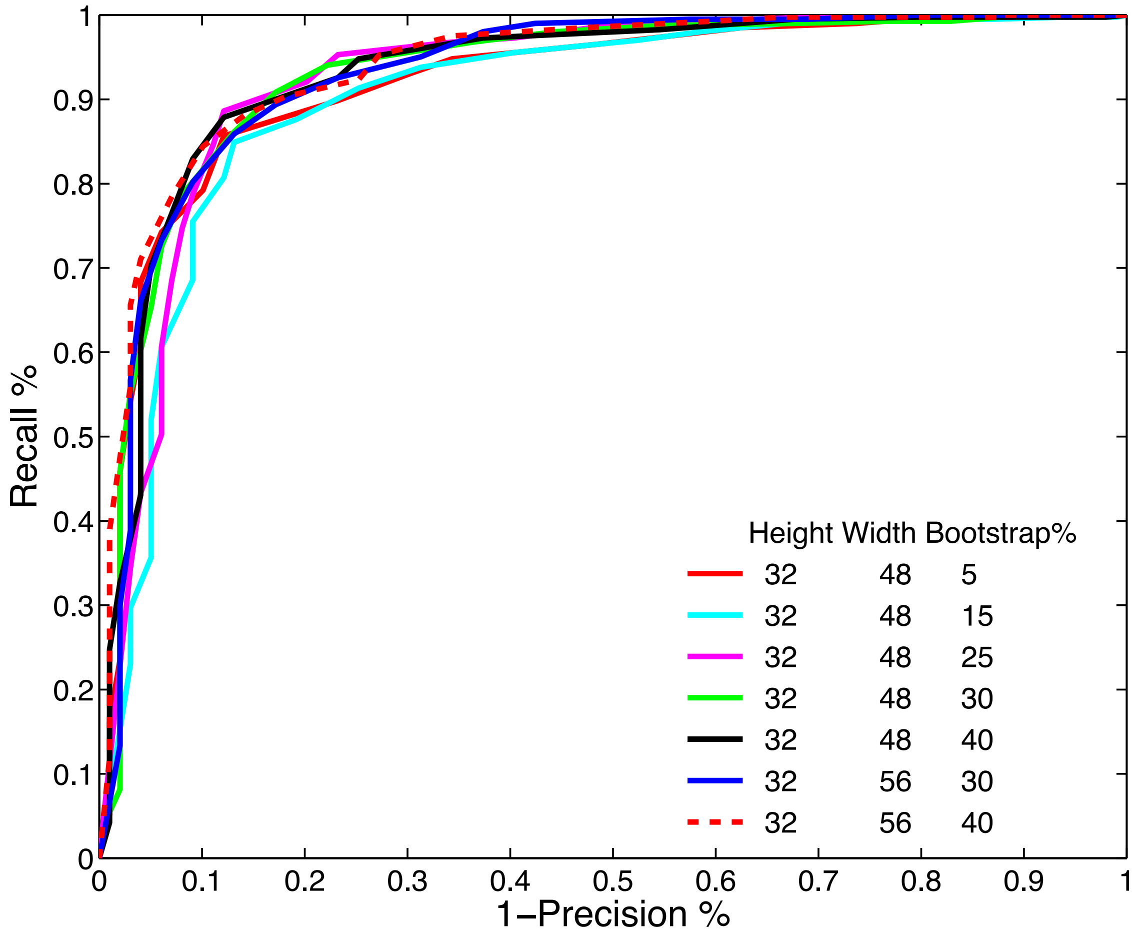

Figure 50. ROC curves for the top seven HOG detectors on 25 percent scale images preformed on GPU.

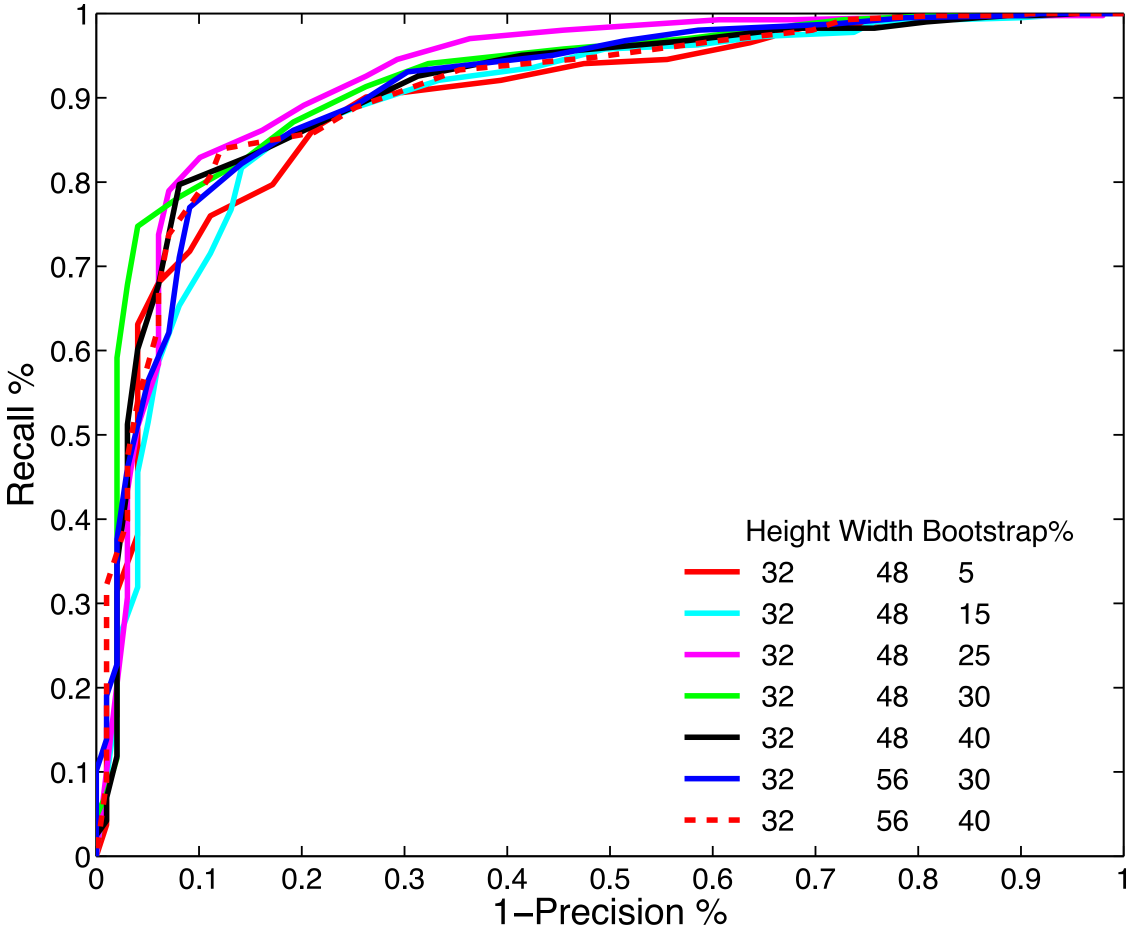

Figure 51. ROC curves for the top seven HOG detectors on 20 percent scale images preformed on GPU.

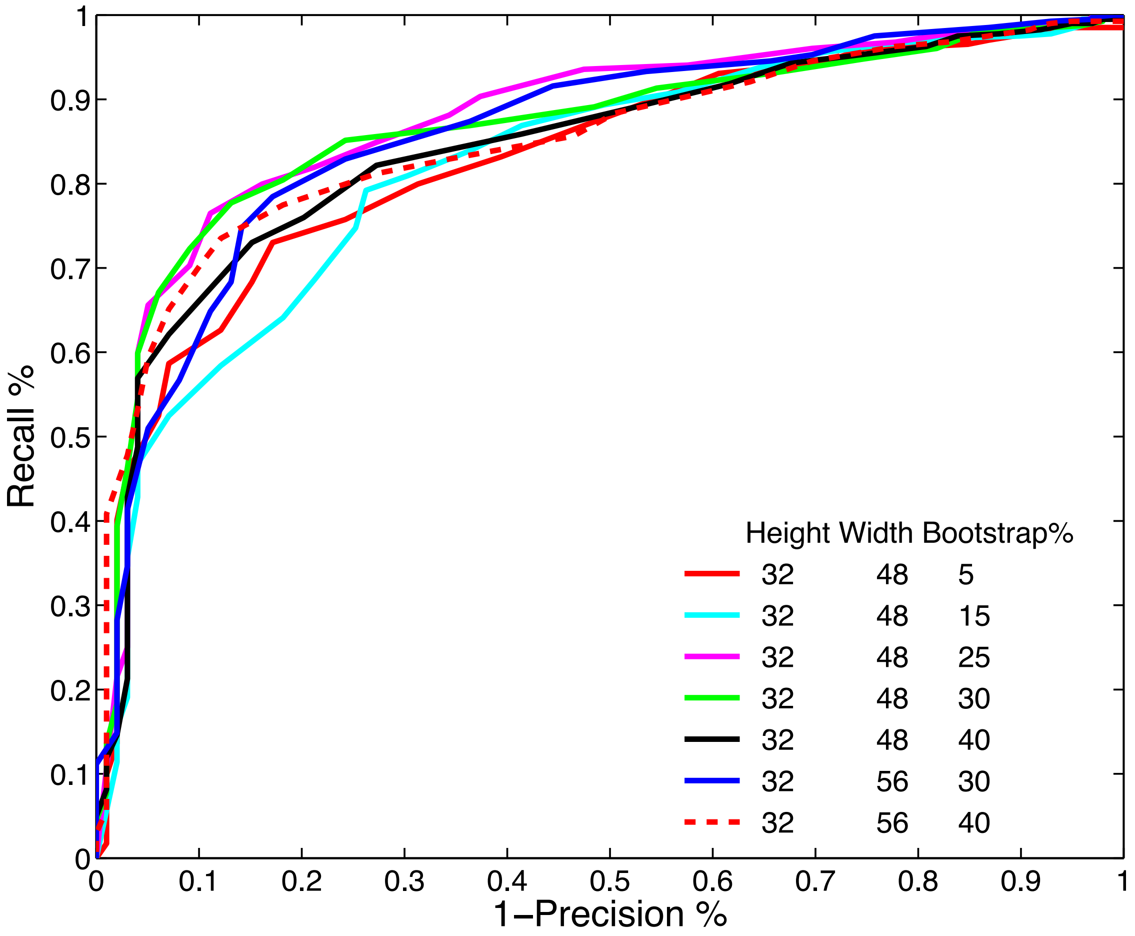

Figure 52. ROC curves for the top seven HOG detectors on 15 percent scale images preformed on GPU.

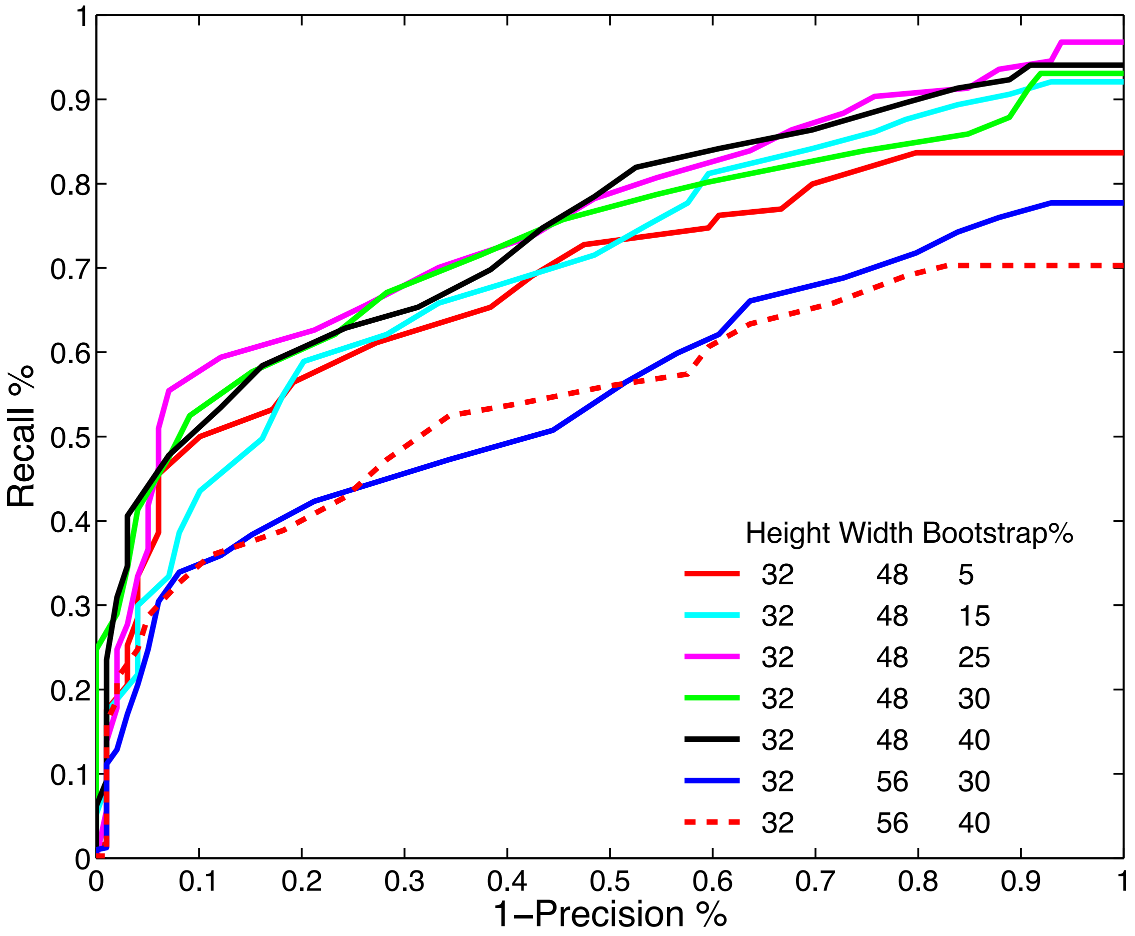

Figure 53. ROC curves for the top seven HOG detectors on 10 percent scale images preformed on GPU.

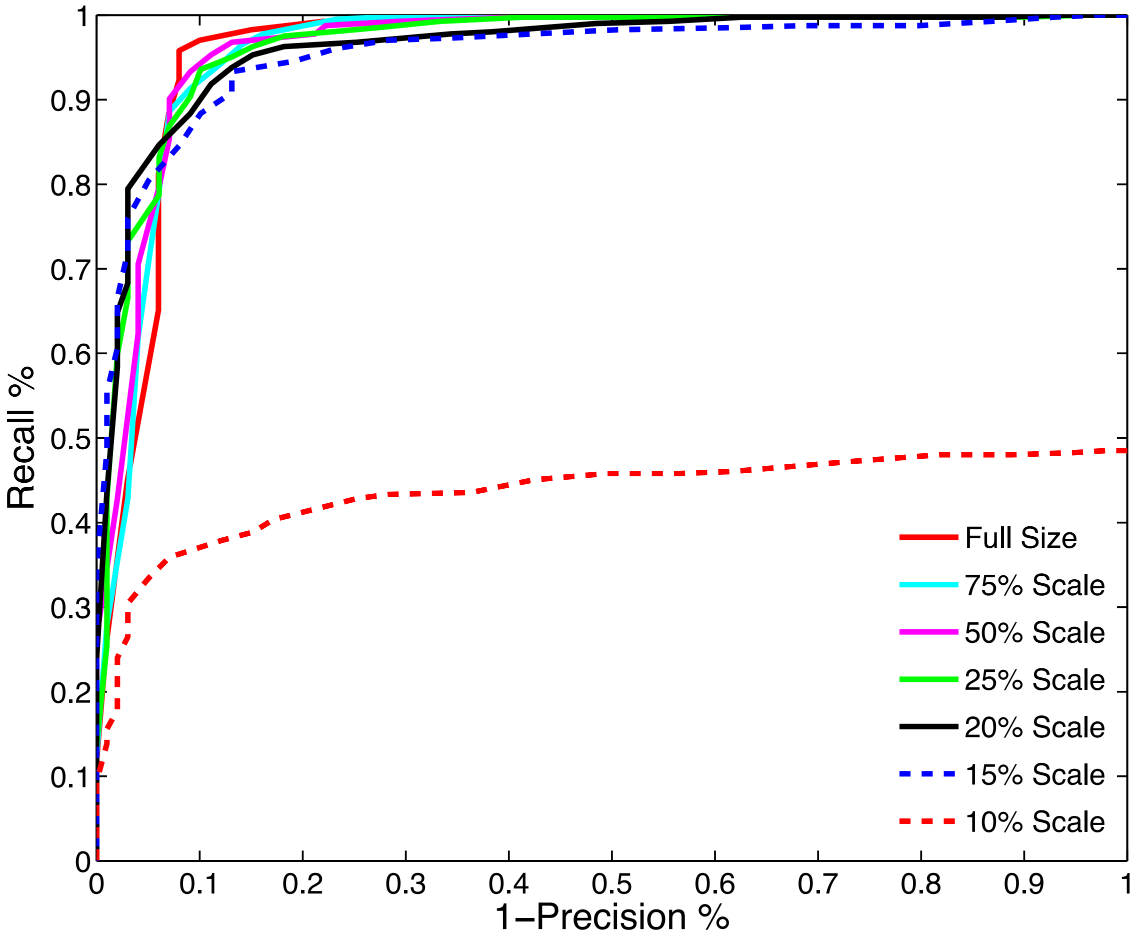

Figure 54. ROC curves for the four-part model DPM detectors on Full size to 10 percent scale images.

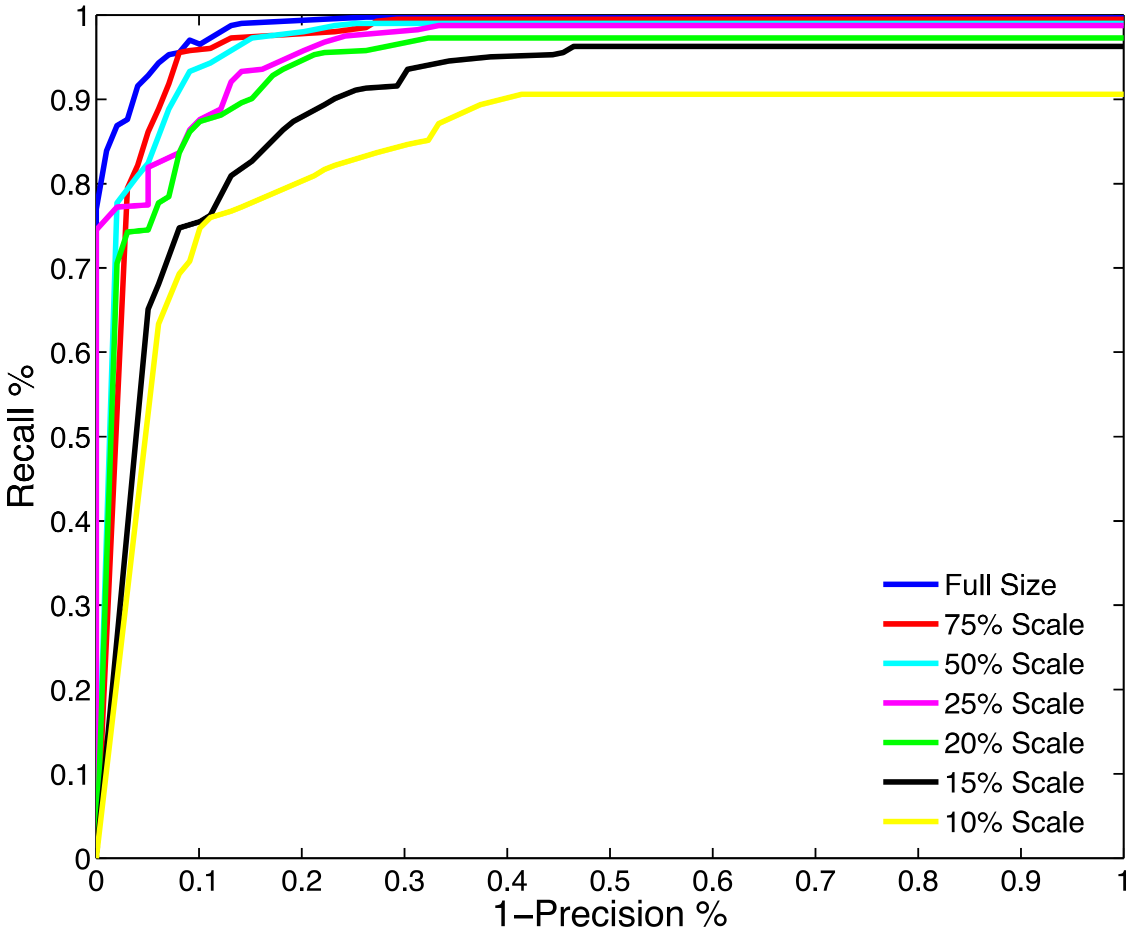

Figure 55. ROC curves for HYBRID HOG and BOW method on full size to 10 percent scale images.

APPENDIX B. LATERAL RANGE CURVES

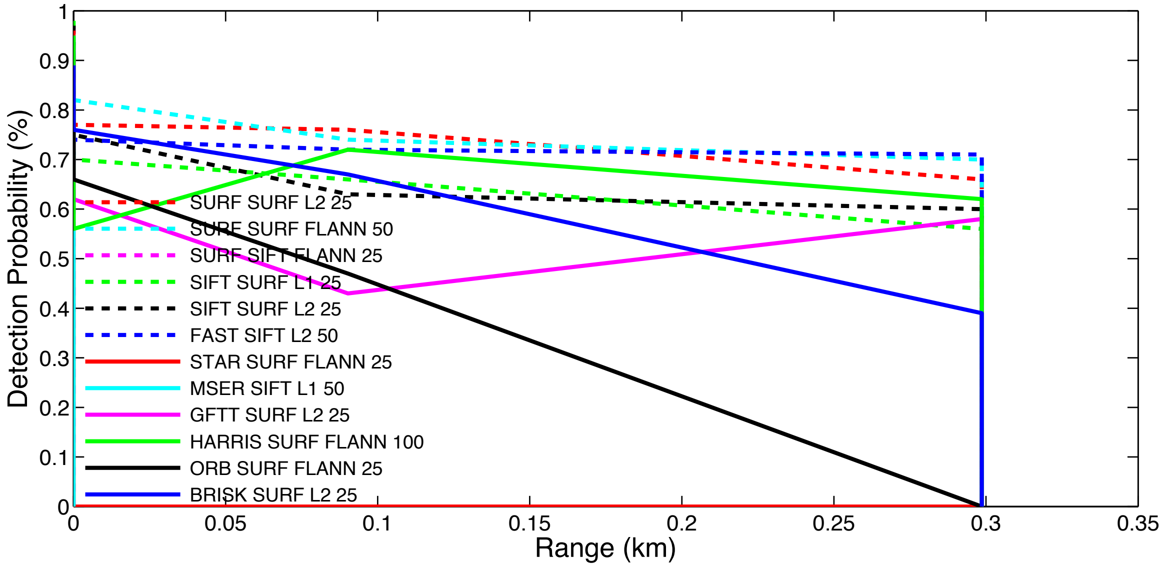

Figure 56. Lateral range curves for the 12 best BOW detectors with WFOV for a 10-meter MHH contact and relative speed of 20 knots.

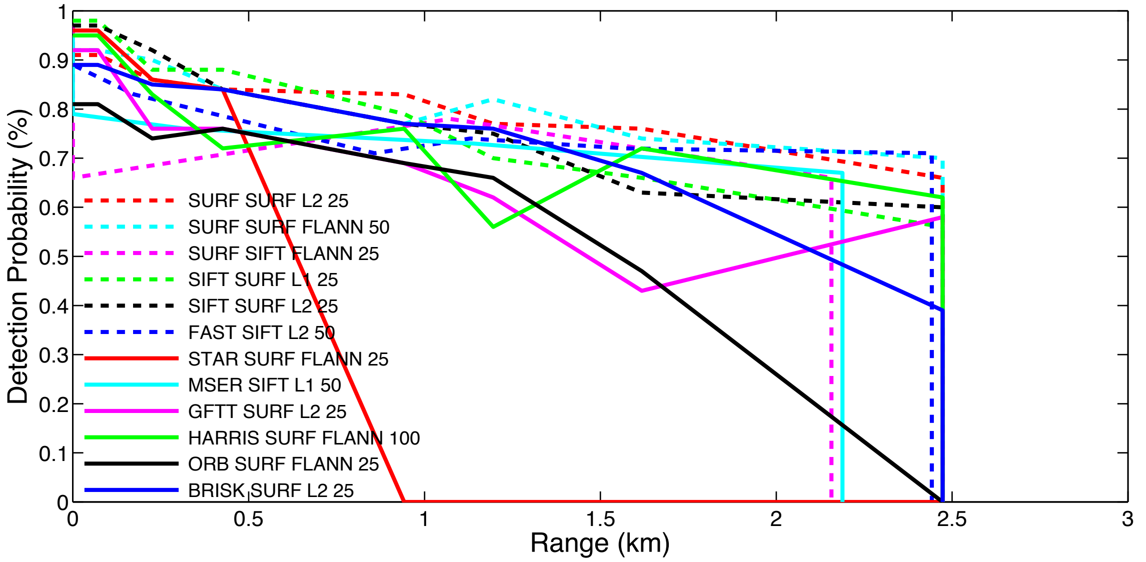

Figure 57. Lateral range curves for the 12 best BOW detectors with MFOV for a 10-meter MHH contact and relative speed of 20 knots.

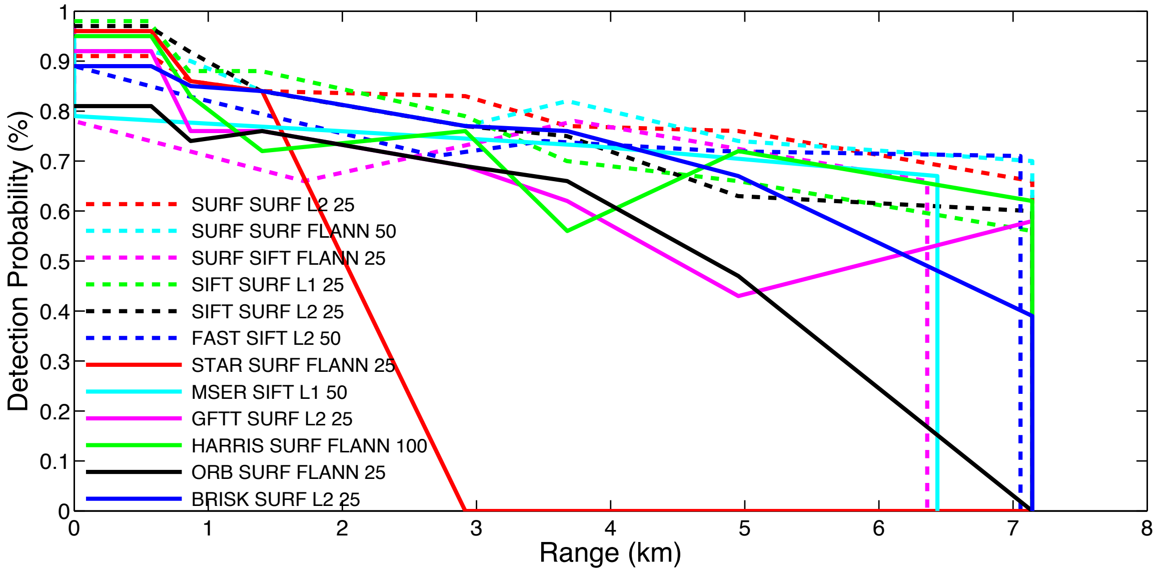

Figure 58. Lateral range curves for the 12 best BOW detectors with NFOV for a 10-meter MHH contact and relative speed of 20 knots.

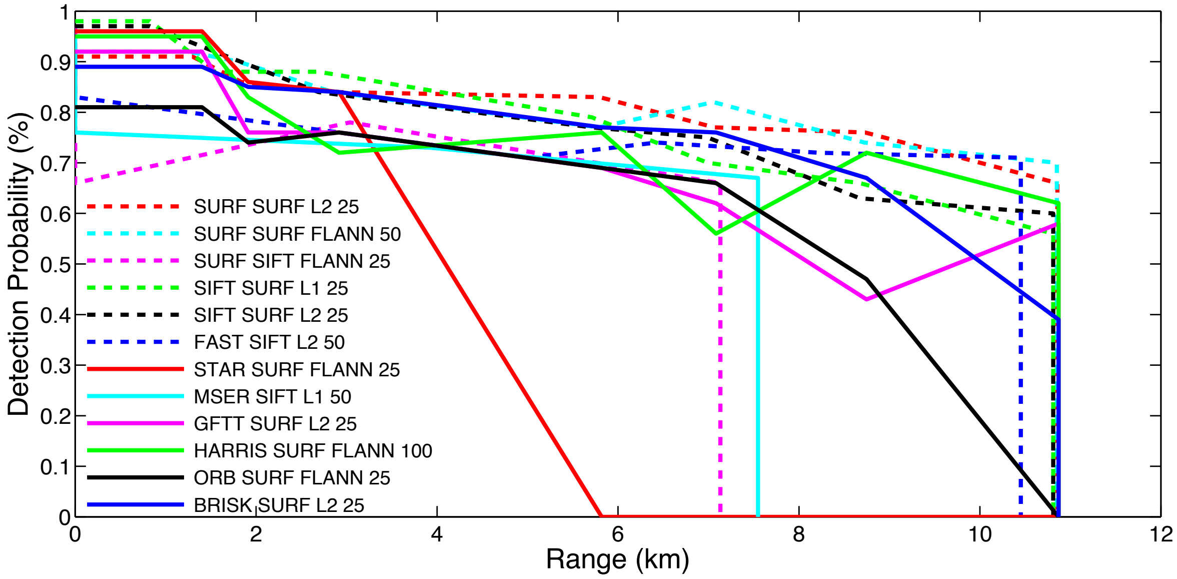

Figure 59. Lateral range curves for the 12 best BOW detectors with UNFOV for a 10-meter MHH contact and relative speed of 20 knots.

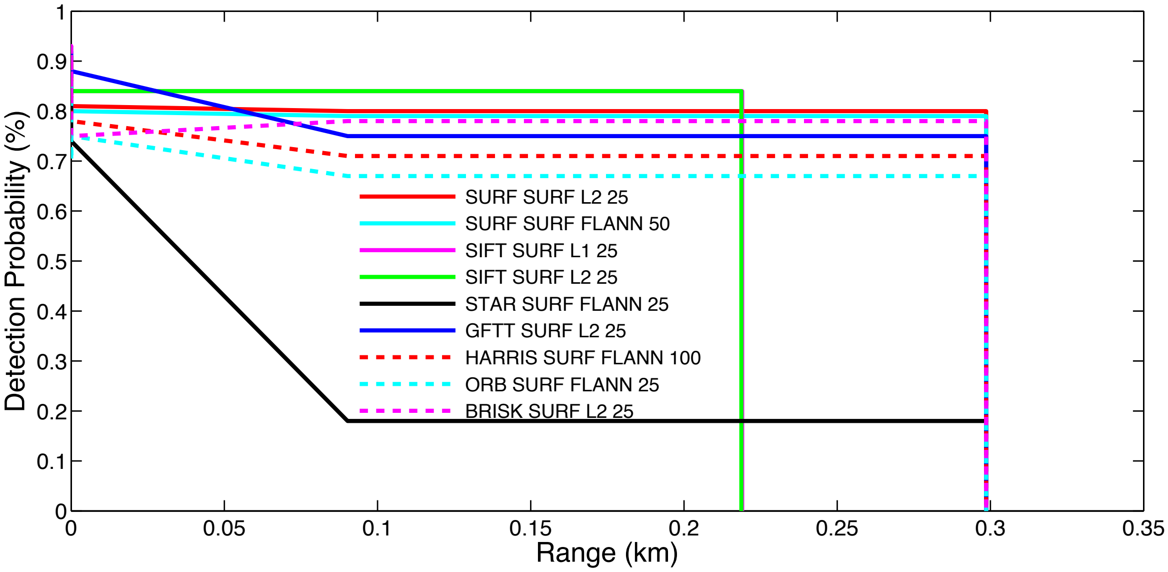

Figure 60. Lateral range curves for the top nine BOW detectors with WFOV, using prescaling of two, for a 10-meter MHH contact and relative speed of 20 knots.

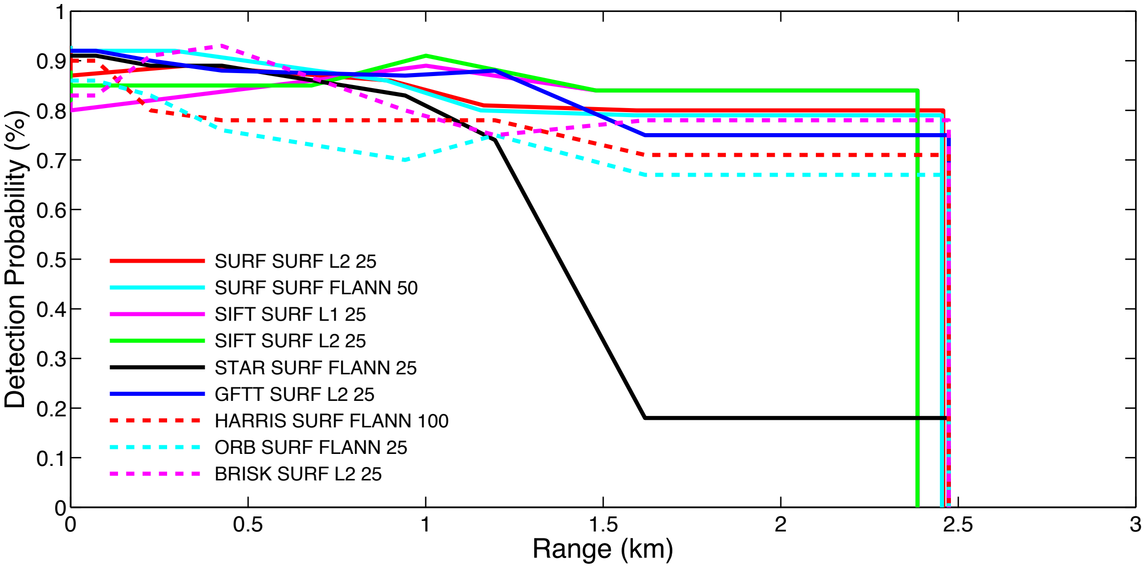

Figure 61. Lateral range curves for the top nine BOW detectors with MFOV, using prescaling of two, for a 10-meter MHH contact and relative speed of 20 knots.

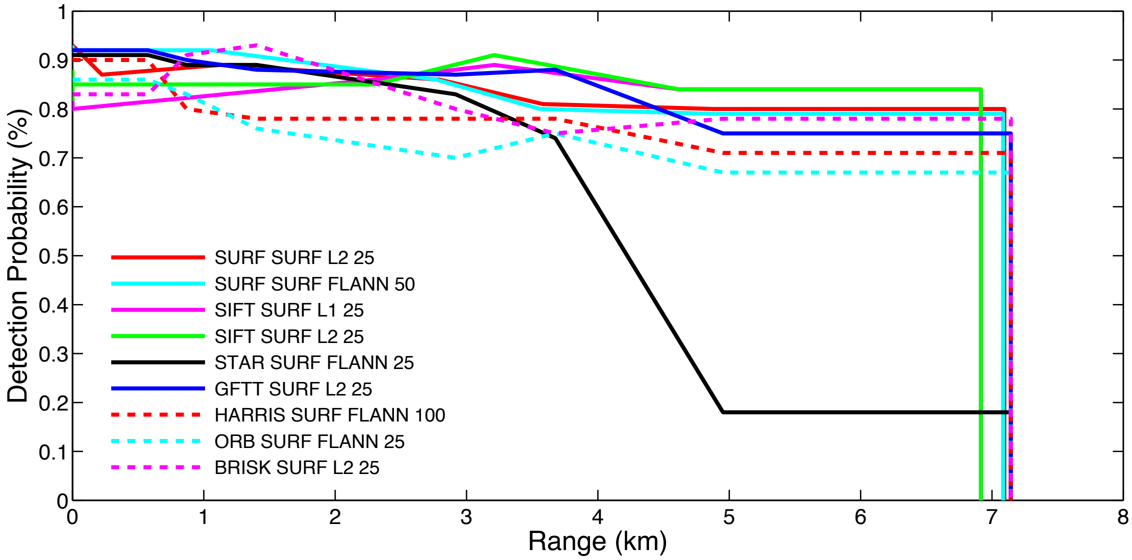

Figure 62. Lateral range curves for the top nine BOW detectors with NFOV, using prescaling of two, for a 10-meter MHH contact and relative speed of 20 knots.

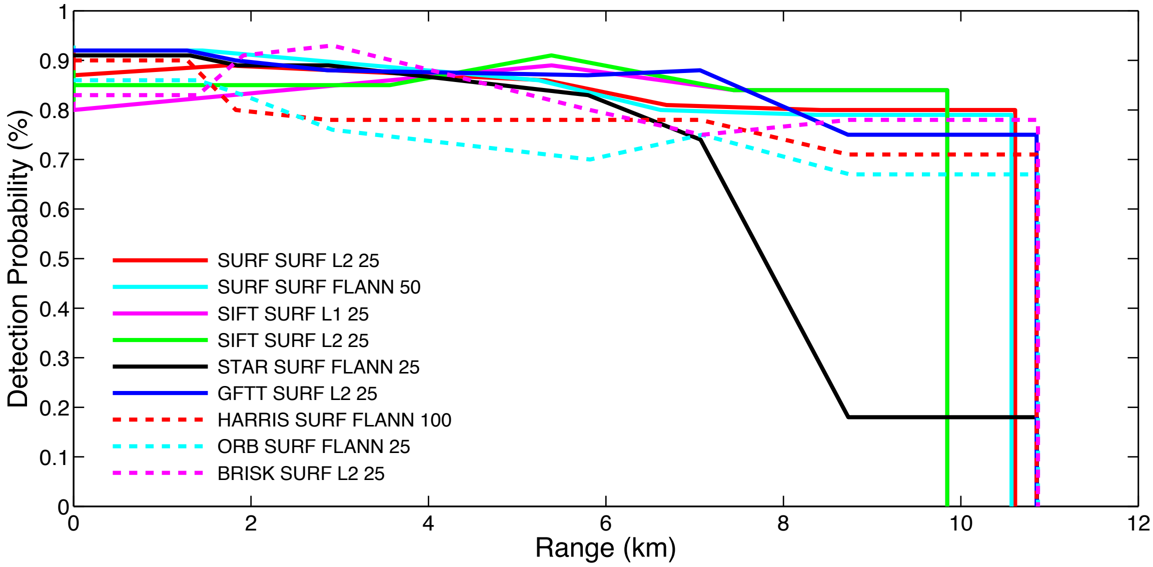

Figure 63. Lateral range curves for the top nine BOW detectors with UNFOV, using prescaling of two, for a 10-meter MHH contact and relative speed of 20 knots.

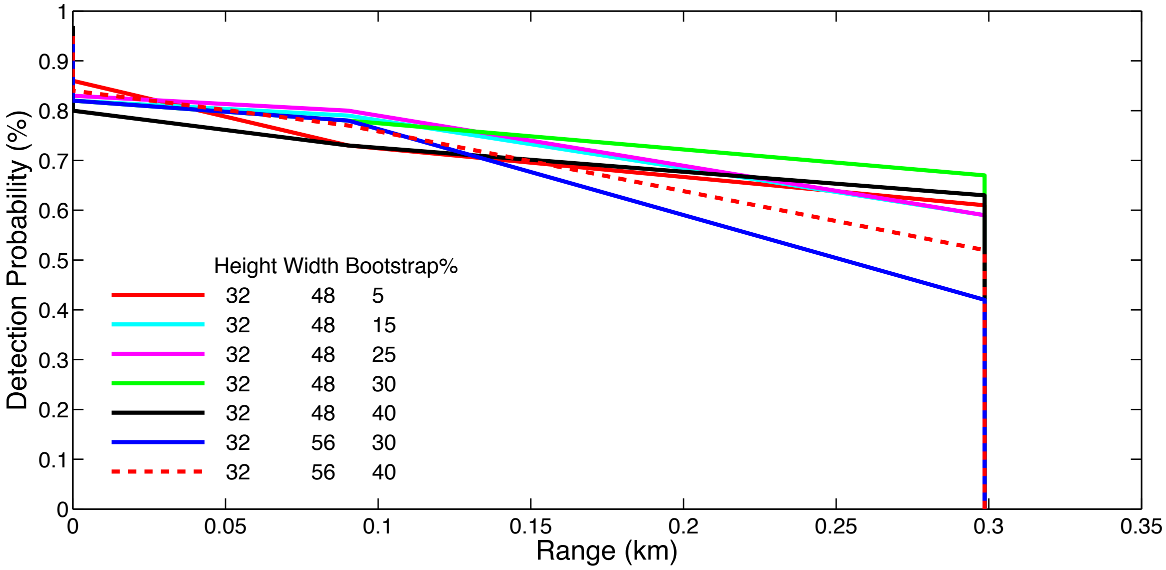

Figure 64. Lateral range curves for the top seven HOG detectors on GPU with WFOV for a 10-meter MHH contact and relative speed of 20 knots.

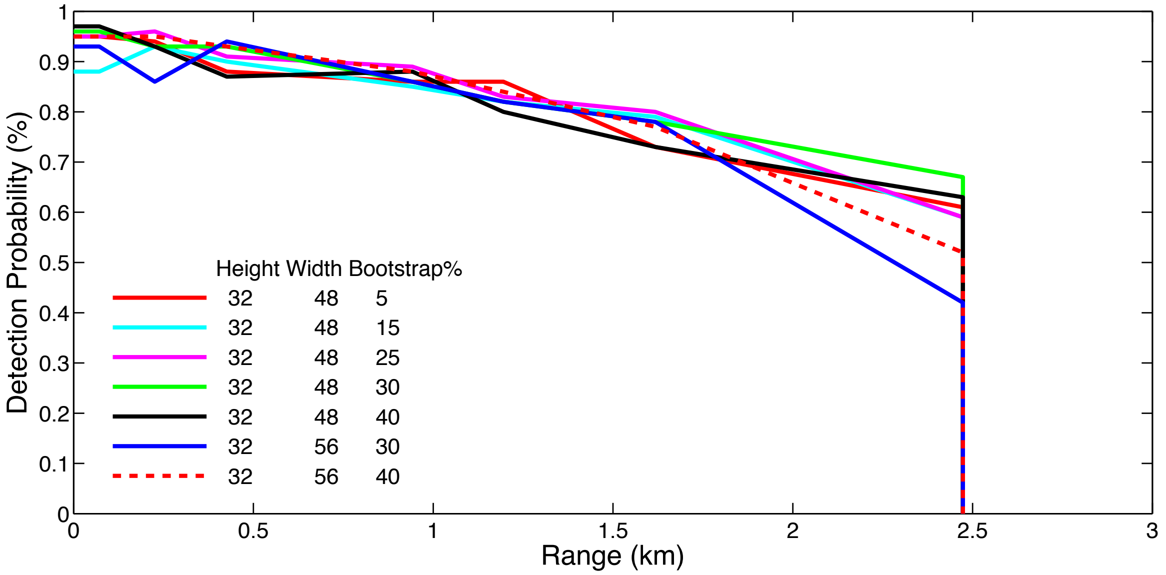

Figure 65. Lateral range curves for the top seven HOG detectors on GPU with MFOV for a 10-meter MHH contact and relative speed of 20 knots.

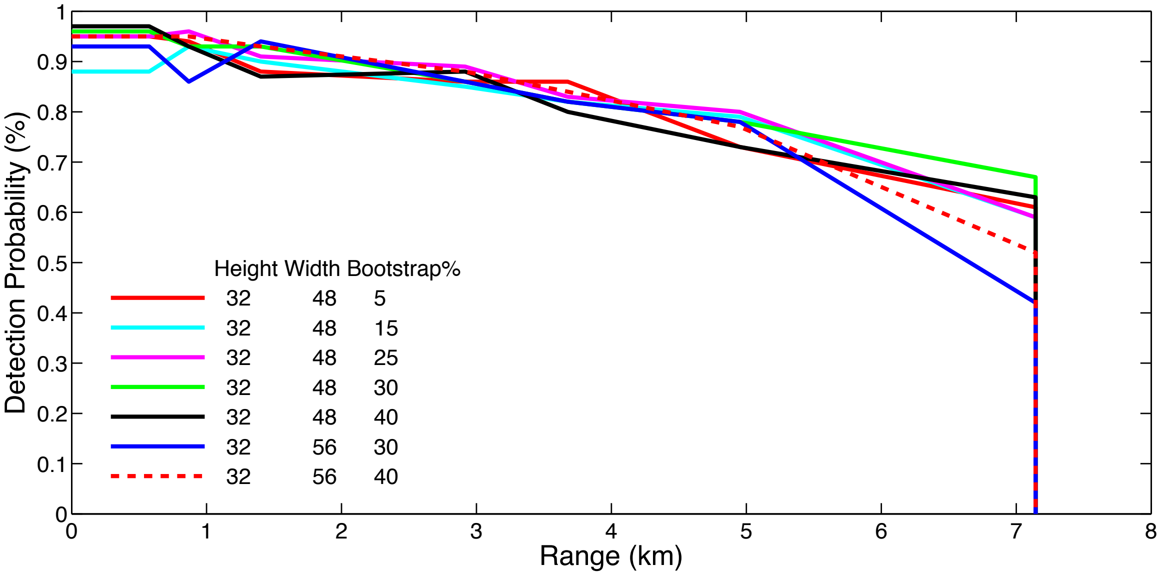

Figure 66. Lateral range curves for the top seven HOG detectors on GPU with NFOV for a 10-meter MHH contact and relative speed of 20 knots.

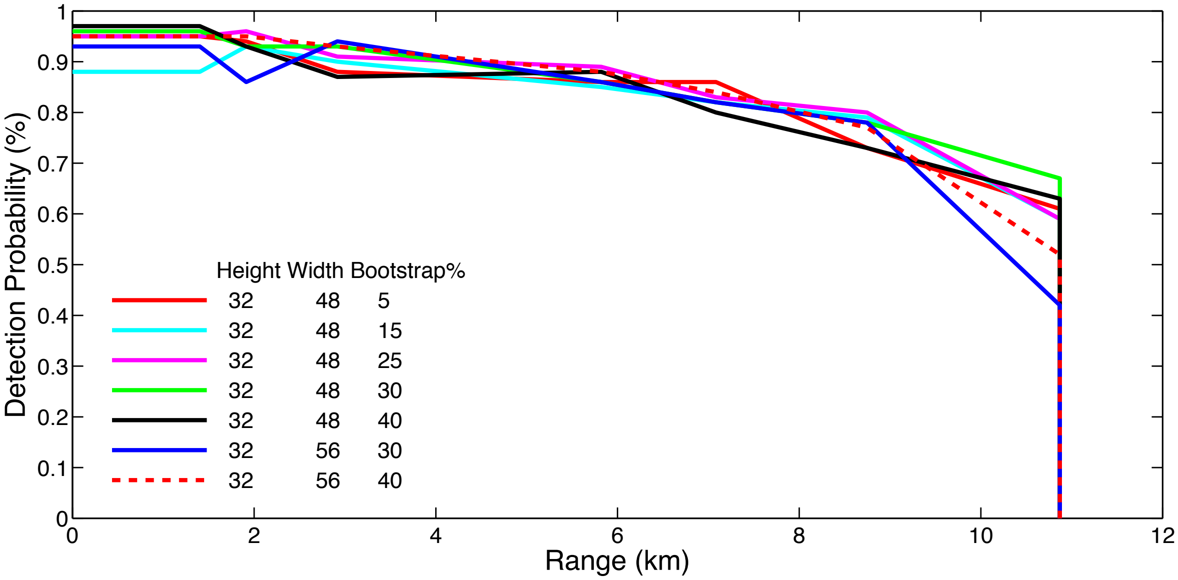

Figure 67. Lateral range curves for the top seven HOG detectors on GPU with UNFOV for a 10-meter MHH contact and relative speed of 20 knots.

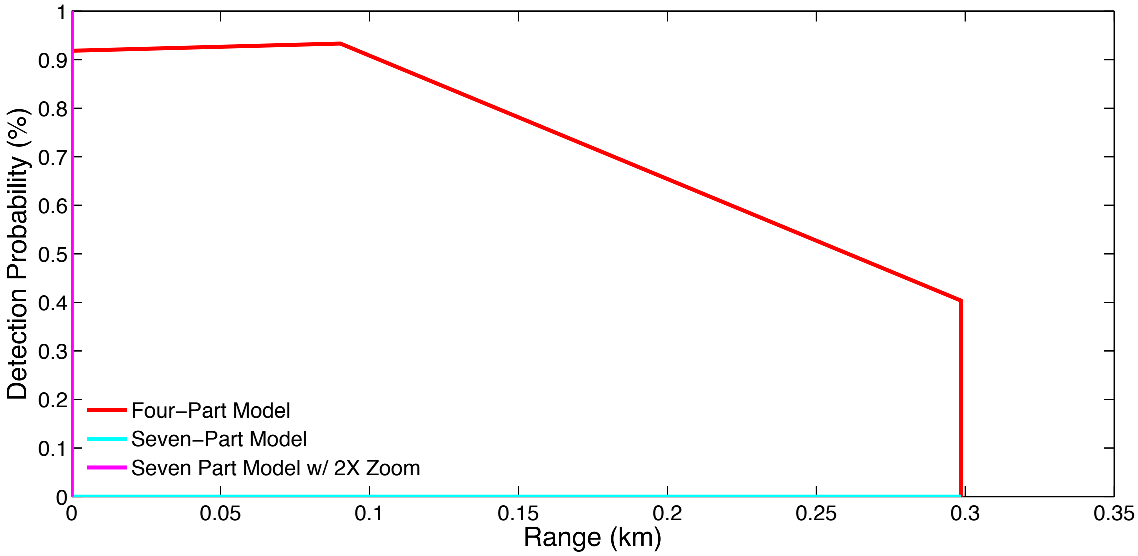

Figure 68. Lateral range curves for the top three DPM detectors with WFOV for a 10-meter MHH contact and relative speed of 20 knots.

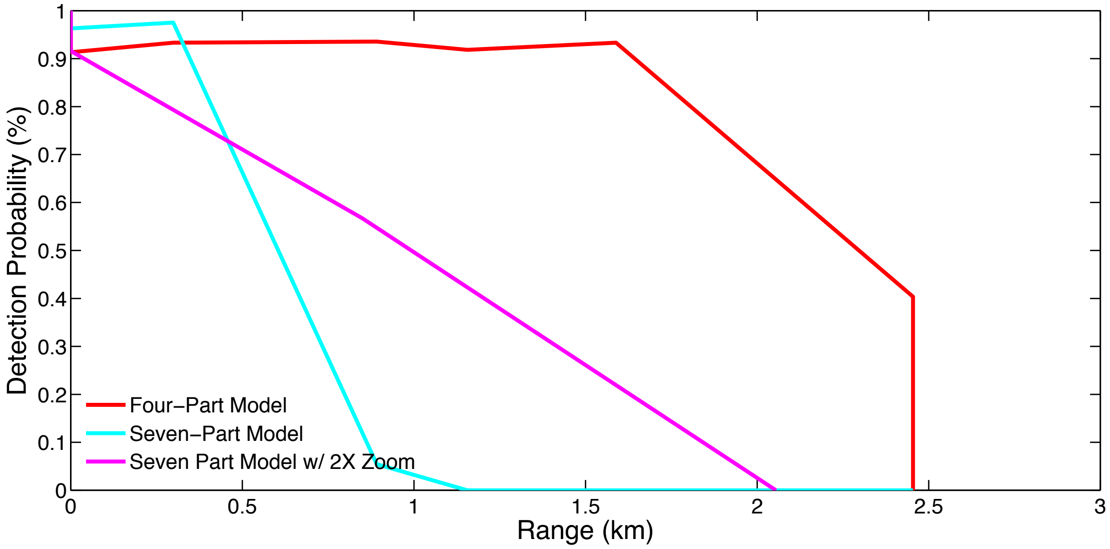

Figure 69. Lateral range curves for the top three DPM detectors with MFOV for a 10-meter MHH contact and relative speed of 20 knots.

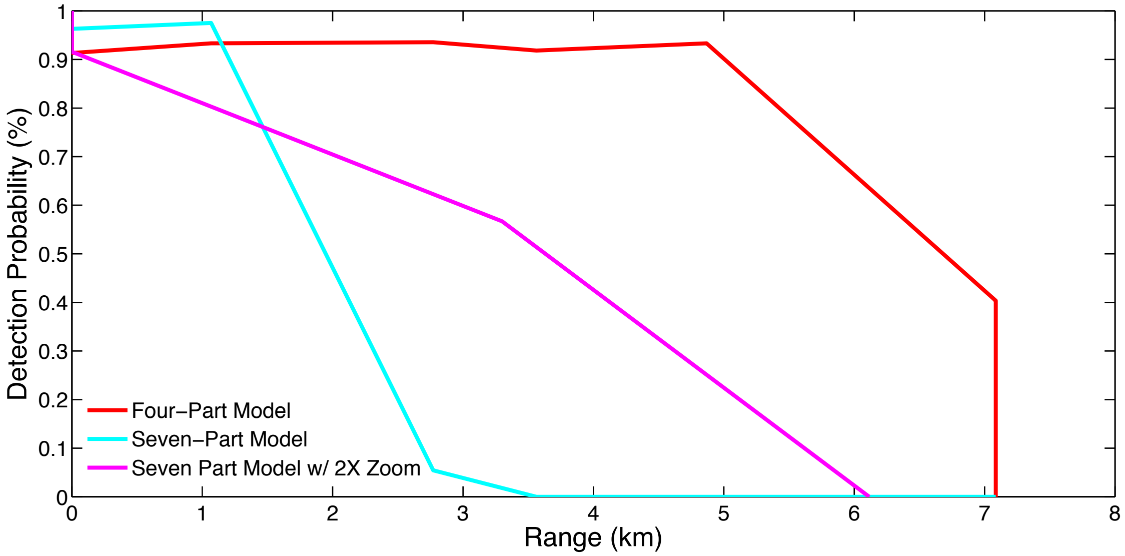

Figure 70. Lateral range curves for the top three DPM detectors with NFOV for a 10-meter MHH contact and relative speed of 20 knots.

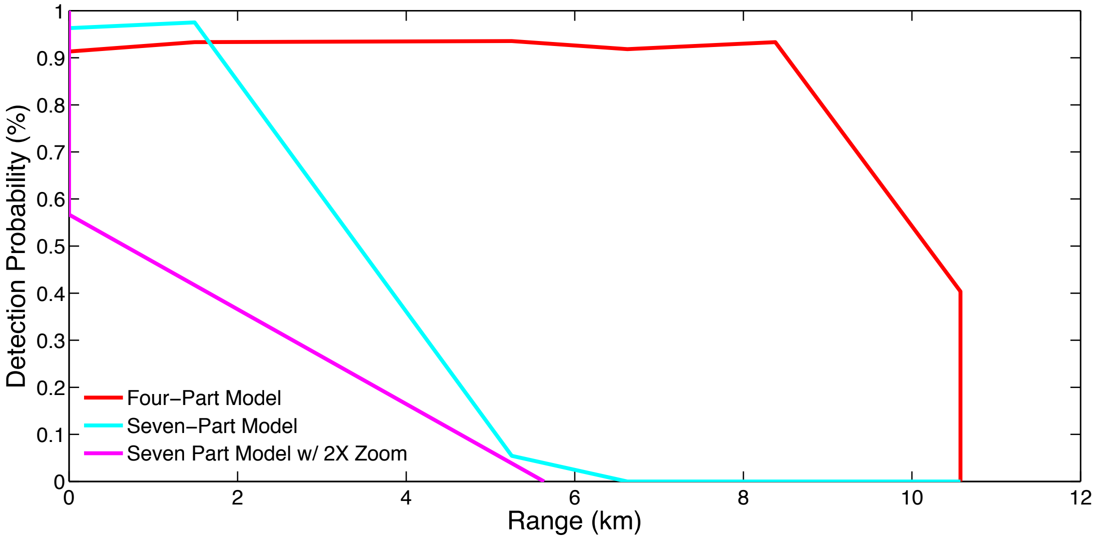

Figure 71. Lateral range curves for the top three DPM detectors with UNFOV for a 10-meter MHH contact and relative speed of 20 knots.

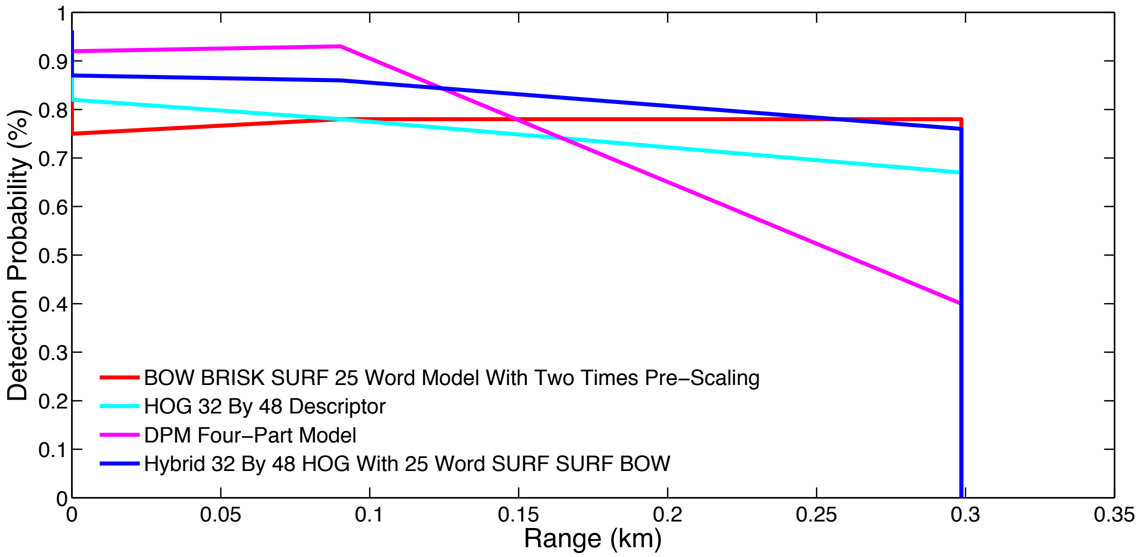

Figure 72. Lateral range curves for the top detectors from each model with WFOV for a 10-meter MHH contact and a relative speed of 20 knots.

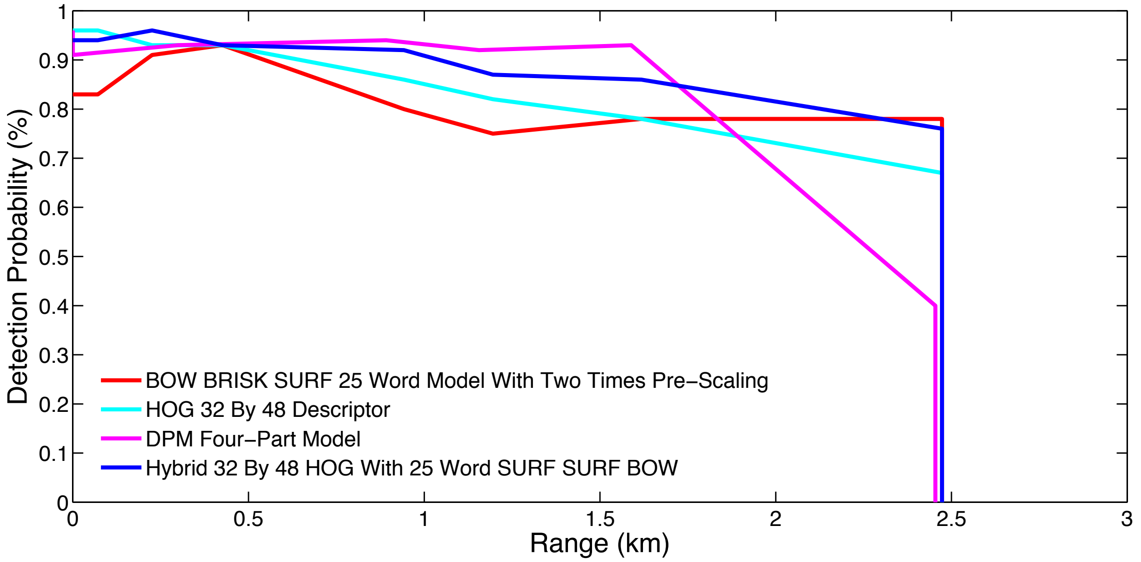

Figure 73. Lateral range curves for the top detectors from each model with MFOV for a 10-meter MHH contact and relative speed of 20 knots.

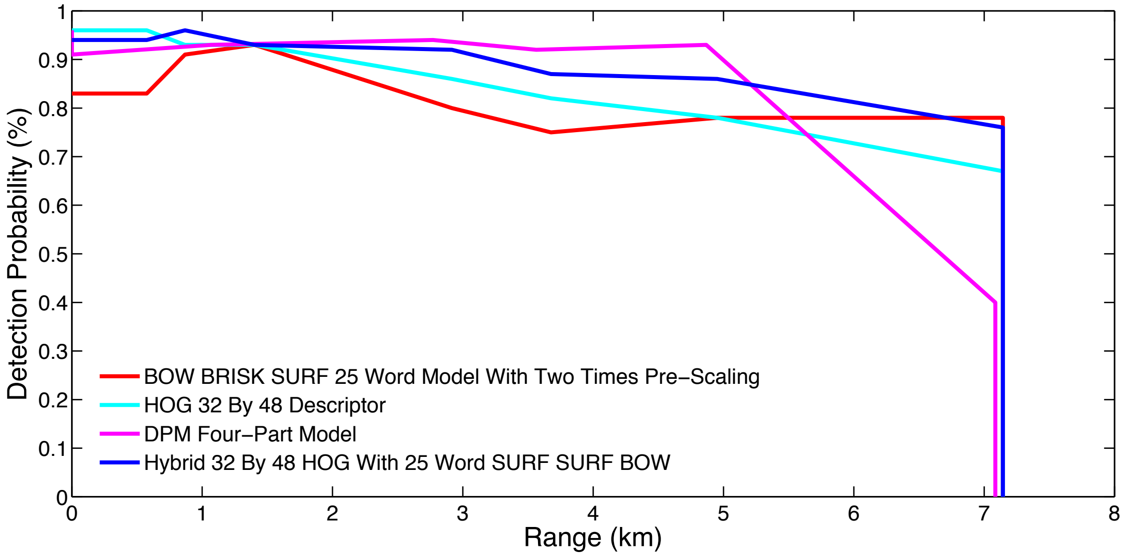

Figure 74. Lateral range curves for the top detectors from each model with NFOV for a 10-meter MHH contact and relative speed of 20 knots.

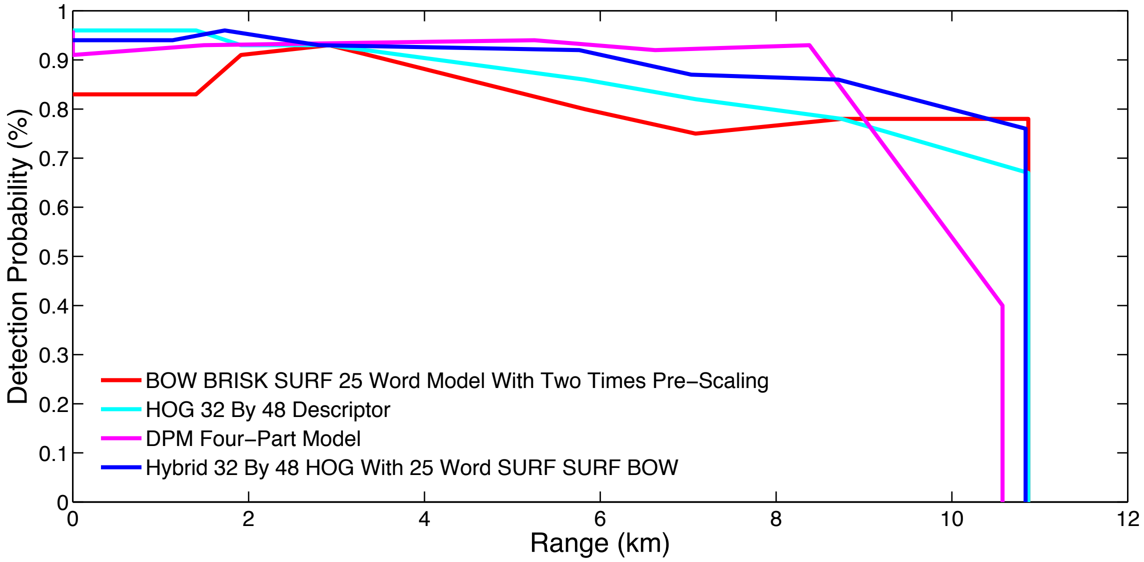

Figure 75. Lateral range curves for the top detectors from each model with UNFOV for a 10-meter MHH contact and relative speed of 20 knots.

LIST OF REFERENCES

[1] K. Rainey et al., “Vessel classification in overhead satellite imagery using learned dictionaries,” *Applicat. of Digital Image Process. XXXV*, SPIE, 84992F, Oct. 2012.

[2] G. Csurka et al., “Visual categoriation with bags of keypoints,” in *Workshop on Statistical Learning in Comput. Vision, ECCV*, Prague, Czech Republic, 2004, pp. 1–22.

[3] P. A. Feineigle et al., “Ship recognition using optical imagery for harbor surveillance,” in *Proc. of Assoc. for Unmanned Vehicle Syst. Int.* (AUVSI), Washinton DC, Aug. 2007, pp. 249–263, vol. 1.

[4] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *IEEE Computer Society Conf. on Comput. Vision and Pattern Recognition*, vol. 1, San Diego, CA, 2005, pp. 886–893.

[5] K. Lillywhite et al., “Real-time human detection using histograms of oriented gradients on a GPU,” in *Workshop on Applicat. of Comput. Vision*, Salt Lake City, UT, 2009, pp. 1–6.

[6] Y. Yang and D. Ramanan, “Articulated pose estimation with flexible mixtures-of-parts,” in *Conf. on Comput. Vision and Pattern Recognition*, Colorado Srpings, CO, 2011, pp. 1385–1392.

[7] M. Everingham et al., “The pascal visual object classes (VOC) challenge,” In *Int. J. of Comput. Vision*, vol. 88, pp. 303–338, 2010.

[8] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” in *Int. J. of Comput. Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[9] S. Zhang et al., “Ship tracking using background subtraction and inter-frame correlation,” in *Int. Conf. on Image and Signal Process*, Tianjin, China, 2009, pp. 1–4.

[10] S. Fefilatyev and D. Goldgof, “Detection and tracking of marine vehicles in video,” *19th Int. Conf.e on Pattern Recognition*, Tampa, FL, 2008, pp. 1–4.

[11] S. Fefilatyev et al., “Tracking ships from fast moving camera through image registration,” in *20th Int. Conf. on Pattern Recognition*, Istanbul, Turkey, 23–26 Aug. 2010, pp. 3500–3503.

[12] S. Fefilatyev et al., “Detection and ttracking of ships in open sea with rapidly moving buoy-mounted camera system,” *Ocean Eng.*, Nov. 2012, pp. 1–12, vol. 54.

[13] M. Everingham et al. (2012). *Pattern analysis, statistical modelling and computational learning visual object classes challnege 2012 results*. [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html

[14] S. Fefilatyev et al., “Horizon detection using machine learning techniques,” in *5th Int. Conf. on Mach. Learning and Applicat.*, Orlando, FL, Dec. 2006, pp. 17, 21.

[15] C. Harris and M. Stephens, “A combined corner and edge detector,” in *Alvey Vision Conf.*, Manchester, United Kingdom, 1988, pp. 147–152.

[16] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” in *9th European Conf. on Comput. Vision*, Graz, Austria, 2006, pp. 430–443.

[17] H. Bay et al., “SURF: speeded up robust features,” in *Comput. Vision and Image Understanding*, 2008, pp. 346–359, vol. 110.

[18] M. Agrawal et al., “CenSurE: center surround extremas for realtime feature detection and matching,” in *Workshops European Conf. on Comput. Vision*, Marseille, Fance, 2008, pp. 102–115.

[19] G. Bradski, “The OpenCV library,” *Dr. Dobb’s Journal of Software Tools*, 2000.

[20] J. Shi and C. Tomasi, “Good Features to Track,” in *9th IEEE Conf. on Comput. Vision and Pattern Recognition*, Seattle, WA, 1994, pp. 593–600.

[21] J. Matas et al., “Robust wide baseline stereo from maximally stable extremal regions,” in *Image and Vision Computing*, no. 10, 2004, pp. 761–767, vol. 22.

[22] M. Calonder et al., “Brief: Binary robust independent elementary features,” in *Conf. on Comput. Vision In European*, vol 6314, Crete, Greece, 2010, pp. 778–792,

[23] E. Rublee et al., “ORB: an efficient alternative to SIFT or SURF,” in *IEEE Int. Conf. on Comput. Vision*, Barcelona, Spain, 2011, pp. 2564–2571.

[24] S. Leutenegger et al., “BRISK: binary robust invariant scalable keypoints,” in *IEEE Int. Conf. on Comput. Vission*, Barcelona, Spain, 2011, pp. 2548–2555.

[25] K. E. A. van de Sande et al., “Color descriptors for object category recognition,” in *European Conf. on Color in Graph., Imaging and Vision*, Barcelona, Spain, 2008, pp. 378–381.

[26] H. Bristow. (2012, October 25) *Deformable parts-based object recognition for OpenCV* [Online]. Available: http://www.willowgarage.com/blog/2012/10/25/deformable-parts-based-object-recognition-open-cv

[27] Y. Yang and D. Ramanan. (2012) *Articulated pose estimation with flexible mixtures of parts*. [Online]. Available: http://www.ics.uci.edu/~dramanan/software/pose/

[28] J. D. M. Rennie and R. Rifkin, "Improving multiclass text classification with the support vector machine," in *Massachusetts Institute of Technology, AI Memo, AIM-2001–026*, 2001.

[29] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Int. Conf. on Comput. Vision Theory and Applicat.*, Lisboa, Portugal, 2009, pp. 331–340.

[30] A. R. Washburn, *Search and Detection, 4th ed.* Linthicum, MD: Institue for Operations Research and the Management Sciences, 2002.

[31] D. H. Wagner et al., Eds., *Naval Operations Analysis 3rd ed.* Annapolis, MD: Naval Institute Press, 1999.

[32] OpenMP Architecture Review Board. (2011, July). *OpenMP application program interface, version 3.1.* [Online]. Available: http://www.openmp.org/mp-documents/OpenMP3.1.pdf

[33] K. Rogozhin. (2012, September 20). *Intel® threading building blocks release notes version 4.2.* [Online]. Available: http://software.intel.com/sites/default/files/tbb-release-notes-4-2.txt

[34] A. Munshi, Ed. (2011, June 1). *The OpenCL specification, version 1.1, Revision 44.* [Online]. Available: http://www.khronos.org/registry/cl/specs/opencl-1.1.pdf

[35] NVIDIA (2013, July 19). *CUDA C programming guide, version 5.5.* [Online]. Available: http://docs.nvidia.com/cuda/cuda-c-programming-guide/

[36] A. Baksheev et al., "Realtime computer vision with opencv," in *Processors*, Apr. 2012.

[37] X. Zhu et al., "Do we need more training data or better models for object detection?," in *British Machine Vision Conf.*, Surrey, United Kingdom, 2012.

[38] J. Ponce et al., "Dataset issues in object recognition," *Toward Category-Levl Object* Recongnition, 2006, pp. 29–48, vol. 4170.

[39] NOAA (2012). National Oceanic and Atmoshperic Administration photo library. [Online]. Available: http://www.photolib.noaa.gov/

[40] IHS (2012). IHS Jane's. [Online]. Available: http://www.janes.com/products/janes/index.aspx

[41] www.shipphotos.co.uk (2013). *www.ShipPhotos.co.uk.* [Online]. Available: http://www.shipphotos.co.uk/hull/

[42] Kodak (2007. March 19). *Kodak KAI-2093 image sensor.* [Online]. Available: http://mpg.ndlab.net/wp-content/uploads/2009/10/KAI-2093LongSpec.pdf

[43] D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Upper Saddle River, NJ: Prentice Hall, 2003.

[44] B. C. Russell et al., "LabelMe: a database and web-based tool for image annotation.," May 2008, pp. 157–173, vol. 77.

[45] A. T. Young (2012). *An introduction to green flashes.* [Online]. Available: http://mintaka.sdsu.edu/GF/explain/atmos_refr/horizon.html

[46] P. M. Roth et al., "Object detection with bootsrapped learning," *Proc. 10th Comput. Vison Winterworkshop*, Zell an der Pram, Austria, 2005, pp. 33–42.

[47] C. Geyer and S. Zickler, "GPU-enabled fast and accurate multi-class object recognition," *GPU Technology Conf. 2014*, San Jose, CA, 2013, p. ID S3102.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California